ROUTING PREDICTION FOR PROBABILISTIC MOBILITY MODEL USING NEURAL NETWORKS FOR AD-HOC NETWORKS

BY

MOHAMMAD RAHIL RAFIQ

A Thesis Presented to the
DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

COMPUTER NETWORKS

MAY, 2010

KING FAHD UNIVERSITY OF PETROLEUM AND MINERALS
DHAHRAN 31261, SAUDI ARABIA
DEANSHIP OF GRADUATE STUDIES

This thesis, written by **Mohammad Rahil Rafiq** under the supervision of his thesis advisor and approved by his thesis committee, has been presented to and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN COMPUTER NETWORKS**.

Thesis Committee

Dr. Tarek Rahil Sheltami
(Advisor)

Dr. Lahouari Ghouti (Member)

Dr. Ashraf S. Hasan Mahmoud
(Member)

Dr. Basem AL Madani
Department Chairman

Dr. Salaam Zummo
Dean of Graduate Studies

2|10|10
Date

DEANSHIP OF GRADUATE STUDIES

Dedicated to my loving Father, Mother, Wife & Sister

ACKNOWLEDGMENTS

In the name of Allah, the Most Beneficent, the Most Merciful

Praise and glory to Almighty Allah (Subhanahu Wa Taalaa) who gave me courage and patience to carry out this work. Peace and blessings of Allah be upon last Prophet Muhammad (Salal La Ho Aleihe Wasallam) and his companions (Radialla-Anhum).

My unrestrained appreciation goes to my advisor, Dr. Tarek Rahil Shaltami, for all the help and support he has given me throughout the course of this work and on several other occasions. I simply cannot begin to imagine how things would have proceeded without his help, his support, and patience. I wish to thank my thesis committee members, especially Dr. Lahouri Ghouti, who helped me not only in the weekdays but also on weekends and Dr. Ashraf S. Hasan Mahmoud for the help and contributions. All made my pursuit of higher education a truly enjoyable and unforgettable experience.

Thanks are due to the chairmen of the Computer Engineering Department Dr. Basem AL-Madani for his support and assistance. Also thanks are due to the faculty members of computer engineering and information and computer science department for their timely help. Special thanks to my dearest mother, father, wife & sister for their emotional support, love, sacrifices, prayers and understanding throughout my academic career. I would not be where I am in life if it weren't for their love.

There are also friends Syed Usama Idress, Farhan Khan and Asif Karim who always provided a very informative, refreshing & a great company. They made me confident of my abilities and helped me tackle towering workloads and solve convoluted problems.

Their constant and sincere encouragement gave me determination to work towards my goal and kept me motivated during hard times. They made all hurdles look like illusions. The presence of Eesaah Lawal, Babar Hasan Khan, Asif Siddiqui, Akhlaq, Salman Ahmed, Zeehasham Rasheed, Saad Rafiq, Syed Faraz, Khaja Wahab, Jamal Khan, Munim Al Mehboob cannot be ignored either, who made my stay in Building 903 as one of the most enjoyable and memorable one.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

THESIS ABSTRACT

Full Name: Mohammad Rahil Rafiq

Thesis Title: Routing prediction for probabilistic mobility model using neural networks for ad-hoc networks

Major Field: Computer Networks

Date of Degree: May, 2010

Ad-hoc network is one of the key areas in wireless networks. There is always room for improvement in order to utilize the available resources optimally. One of the thought among these was to use the prediction algorithms to predict the next route of the mobile nodes which can help us in allocating the resources in advance and also controlling the network traffic. So this thesis mainly emphasis on the analysis of the behavior of AI based Routing Protocol for Adhoc networks suitable for video traffic Research has been done in this area using Neural Networks but no typical model was designed for predicting the routes due to lack of standard dataset to train the model. We generated different datasets using Gaussian Markov mobility model which has some concrete reasons or benchmarks to evaluate the end products' performance. Artificial Neural Networks (NN) & Extreme learning machine (ELM) were used to build the prediction models.

ملخص الرسالة

الاسم الكامل : محمد رحيل رفيق

عنوان الرسالة : التوجيه التنبؤ لنموذج احتمالي التنقل باستخدام الشبكات العصبية المخصصة للشبكات

مجال التخصص : شبكات الحاسب

تاريخ الدرجة العلمية:مايو 2010

شبكة المخصص هي واحدة من المجالات الرئيسية في الشبكات اللاسلكية . هناك دائما مجالا للتحسين من أجل الاستفادة من الموارد المتاحة على النحو الأمثل. وكان من بين هذه الفكرة لاستخدام خوارزميات التنبؤ التنبؤ الطريق القادم من العقد المتنقلة التي يمكن أن تساعدنا في تخصيص الموارد في وقت مبكر والسيطرة على حركة مرور الشبكة أيضا . حتى هذه الأطروحة التركيز أساسا على تحليل سلوك التوجيه منظمة العفو الدولية القائمة على بروتوكول مناسب للشبكات مخصصا للفيديو حركة البحث تم القيام به في هذا المجال باستخدام الشبكات العصبية ولكن لا يوجد نموذج تم تصميمه النموذجي للتنبؤ الطرق بسبب عدم وجود معيار لمجموعة البيانات قطار النموذج. إنشاء قواعد البيانات المختلفة باستخدام نحن غاوسي نموذج ماركوف التنقل التي لديها بعض أسباب محددة أو معايير لتقييم أداء المنتجات النهائية . تم استخدام الشبكات العصبية الصناعية (NN) والمتطرفة تعلم آلة (ELM) لبناء نماذج التنبؤ.

CHAPTER 1

INTRODUCTION

Wireless networks have a major impact on the world with which information could be sent to multiple senders and receivers efficiently and more reliably. Wireless networks have continued to develop and their usage has grown significantly such as cellular phones are part of huge wireless network systems. Other than cellular network, wireless network have applications for small areas which includes ad-hoc networks and wireless sensor networks. Besides all these, the demand of the users for better quality of service and elevated performance is persistently mounting, and requires improving the utilization of resources. One of the most important issues in the mobile networks is the location management. Location management is the process by which the current location of a mobile host is determined. It can be further subdivided into two different services; mobile tracking and locating. Mobile tracking is the process by which the network keeps track of the current location of the mobile host, whereas Mobile locating deals with the process of finding the current location of the mobile host for the delivery of an incoming call [10].

One way of searching a mobile node to identify its current location is to broadcast search information, called paging, in each and every cell in the whole geographical area. However, the amount of channel bandwidth consumed by these numerous broadcast signals can be extremely high. The other way of searching is to store the current location of each mobile host by a location update signal from the mobile hosts and maintaining the location information database for each mobile host [10], which is more expensive and has high signaling traffic. One of the methods is to use mobile host movement behavior and their traffic characteristics to predict the future location of a mobile host. If the location of an mobile node is known in advance, the overhead will be avoided. Thus, by maintaining the knowledge of the mobile host history of movement behavior, the signaling overload can be reduced. By using the prediction model, we intend to reduce the number of hello messages that are sent by the nodes among themselves after every specific time interval so that the nodes within the network are able to update their routing tables or neighbor list. In our case, instead of using the conventional hello messages, our algorithm will predict the next location of the mobile nodes and the nodes will update the routing tables or neighbor list locally.

1.1 Wireless Sensor Networks

A wireless sensor network (WSN) is a wireless network related to distributed autonomous devices using sensors to courteously monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants, at different locations. They are small, inexpensive, low-power, intelligent, disposable sensors can be deployed in large numbers deployed manually by hand, or randomly by, for instance, dropping them from an aero plane. These sensor nodes are self configuring and include one or more sensors, embedded wireless communications and data processing components. Besides one or more sensors, each node in a sensor network is usually equipped with a radio transceiver or other wireless communications device, a small microcontroller, and an energy source, generally a battery. However, WSN are used in many industrial application areas, such as industrial process, control, machine health, environment and habitat monitoring, etc. The key challenge of WSN is energy efficiency. The energy of a node is worn out by:

i. Computational processing and
ii. Data transmission and receiving

Both of these factors are controlled by the network layer. An efficient network layer can reduce the number of messages that are sent by a node as well as the complexity of the computation of routing paths. Routing protocols can be subdivided into source initiated and destination initiated protocols. Nodes which uses source initiated protocols send information either periodically, or in response to certain events such as polling in their environment. In destination initiated routing, nodes only send data in response to a request for data. The shortcoming of destination initiated protocols is the fact that requests are usually flooded through the network, which results in exhausting the energy sources of nodes [3].

1.2 Ad-hoc network

Wireless communication between mobile users has become more popular nowadays than ever before. This is due to the recent advances in wireless communications technologies combined with the introduction of small portable computing devices, which are the two main reasons why mobile computing continues to enjoy rapid growth. An ad-hoc network is a concept of computer communications, which means that users waiting to communicate with each other form a temporary network without any form of centralized administration. One feature of ad-hoc networks is that the network will not collapse just because one of the mobile nodes moves out of the transmitting range of the other nodes. Nodes should be able to enter or leave the network as they desire. In others words, ad-hoc networks are capable of managing the changes in topology and breakdown in nodes. In ad-hoc networks, each host must act as a router. Due to continuous movement of the nodes, the backbone of the network is continuously reconstructed. Moreover, the nonexistence of a centralized authority also complicates the problem of medium access. Some of the main restrictions of Ad-hoc network include:

Dynamic topology: As mentioned earlier in the introductory part, ad-hoc networks have very flexible mobility meaning that nodes join and leave the network in a random and dynamic fashion. This results in dynamic links establishment and removal. Hence, links are also subject to frequent disconnection during node's mobility.

Bandwidth constrained: This is not a particular ad-hoc network restriction. In fact, wireless links have significantly lower capacity than the wired links; they are affected by several error sources (interference, fading, environmental condition etc) that result in degradation of the received signal and high bit error rate in the range of 10^{-4} and 10^{-5}.

Energy constrained: The prime focus of the mobile world is not just to improve its performance. If the mobile devices have batteries with more power, this in turn, is hazardous to man's health. So there should be a tradeoff in performance and power. Hence, the most important system design criteria for optimization may be energy conservation.

Limited physical security: Mobility implies higher security risks than static operation because portable devices may be stolen or their traffic may cross insecure wireless links. Eavesdropping, spoofing and denial-of-service attacks are relatively easy to deploy in wireless environment and should be taken care of

Network size: As the network grows, routing delay becomes bottleneck for the network size. This constraint is more significant for video data which are more sensitive to delay than any other traffic. Therefore, there is practical limit on the size of the network or we have to trade off performance or quality of service (QoS) requirements to increase the network size.

Multiple routes: To reduce the number of reactions to topological changes and congestion, multiple routes could be used. If one route has become invalid, it is possible that another stored route could still be valid and thus enhance the robustness of the routing protocol.

Load balancing: The protocol should be distributed and balances the load amongst all the nodes in the network. It should not be dependent on certain nodes to do the routing. All the nodes should participate in the routing function. This enlarges the life of the network and reduces the reduction of connectivity problem.

1.3 Differences between WSNs and MANETS

Wireless sensor network (WSN) differ from other mobile Adhoc networks (MANET) in seven areas, namely network size, node density, node proneness to failure, frequency of topology changes, communication paradigm employed, resource limitations of nodes and node identification [4][5] and these are discussed below.

The size of a WSN can be anything from a few nodes up to many thousands of nodes whereas MANETs generally consist of less than a hundred nodes. Bluetooth, which can consist of up to a maximum of eight nodes, is an example of a MANET. A wireless local area network (WLAN) is another example of a MANET. WLAN is an IEEE 802.11b standard. The size of a WLAN is limited to 32 nodes per access point [2].

Node density in a WSN is usually high, with a large number of nodes in a relatively small area, while other MANETs mostly consist of only a few nodes in close proximity of each other. This is due to the nodes density. A WSN node can be as small as a one Euro coin, while nodes of other MANETs are mostly notebook computers, palmtops or cellular telephones.

A WSN is deployed in a remote area. In such circumstances the node proneness to failure is high due to the possibility of nodes being damaged and failing. Some nodes might also drain their energy resources quicker than other nodes due to being on a routing path that is utilized more than other paths. Nodes in other MANETs have rechargeable energy supplies and are not subjected to adverse environmental conditions that could damage them to the extent of not being able to function any longer.

The frequency of topology changes in a WSN is high, due to factors such as node failures, node additions, nodes moving and environmental interference. The network has to be able to adapt to these changes in node position and number. Topology changes can happen as frequently as every few milliseconds. In other MANETs, nodes usually request to join the network and leave the network after a certain period of time, which is rarely less than a couple of minutes.

The communication paradigm employed in WSNs includes a large number of broadcasts that are sent through the network. These broadcasts are used for network set up and maintenance, discovery of neighbors and sending of data. Other MANETs usually use point-to-point communications, since the source knows how to reach the destination.

The resource limitations of nodes in WSNs include limited energy and bandwidth resources, compared to other MANETs. The energy resources of WSN nodes cannot be replenished, while other MANET's nodes have rechargeable batteries. The limited data rate of up to a few kilobits per second in WSNs is small compared to data rates of between one and a few hundred megabits per second in other MANETs. The memory of WSN nodes is limited to a few kilobytes, while other MANET's nodes can have gigabytes of memory. The processors employed in WSN nodes are limited.

Node identification by means of globally unique identifiers are not always possible in WSNs, due to the possibly very large number of nodes in the network and the overhead caused by having a unique identifier for each node. In other MANETs, the nodes have unique IP addresses.

Sensors can generate a measurable response to changes in physical conditions, such as temperature or areas of magnetic field. The main aim of such networks is to execute distributed sensing tasks, mainly for applications such as environmental monitoring, smart spaces and medical systems.

1.4 H.263 Standard

H.263 is a video compression standardized by International Telecommunication Union (ITU) that appeared in February 1995. Basically, it is designed for low bit-rate communications. It is almost similar to its predecessor standard, H261, but with some enhanced features. Furthermore, five resolutions are supported by the new standard H263.

i. CIF - Common Intermediate Format uses 352 pixels by 288 lines.
ii. QCIF - Quarter Common Intermediate Format uses 176 pixels by 144 lines.
iii. SQCIF - Sub-Quarter Common Interchange Format uses704x576 pixels.
iv. 4CIF - 4 times Common Intermediate Format uses704x576 pixels.
v. 16CIF - 16 times Common Interchange Format uses1408x1152 pixels.

However, if there is significant motion, the H263 gives poor performance as compared to H.261. Hence, it is better for applications such as videoconferencing, where there is little motion. Summarizing its application, we can list the followings [60]:

i. Video over the Internet and over telephone lines

ii. Surveillance and monitoring

iii. Tele-medicine (medical consultation and diagnosis at a distance)

iv. Computer-based training and education

The Discrete Cosine Transform (DCT) exploits spatial redundancy while motion compensation makes use of temporal redundancy. Moreover, features such as half-pixel accuracy, bidirectional coded macro blocks (16x16 and 8x8 for luminance and chrominance respectively) really make H.263 a best fit for low bit rate applications. We tested Quarter Common Intermediate Format (QCIF) with a frame rate of 30 fps (frames per second) in our simulation. Ideally, the compression technique should not exceed frames (frame size) of the delay. But most systems have several layers of buffering and queuing so this delay can even be as large as 10 seconds in some cases. More details on H263 can be found in [40].

1.5 Scope

So the accurate determination of user's future location and the amount of available resources that could be reserved for him is likely to become a very important issue for the future wireless networks. This is also closely related to emerging location aware and context sensitive services for being able to determine the user's future location as he moves inside the network while being connected can result in significant improvement of system efficiency and connection quality. So there has been an increasing demand to develop predictive algorithms for user movements in order to provide better optimization in terms of reducing the routing overheads.

1.6 Objectives

The objective of this research is to analyze the behavior of artificial intelligence (AI) based routing protocol for ad-hoc networks suitable for video traffic. For this purpose we have to develop models for next-move prediction scheme using the history of the user's movement directions using artificial intelligence (AI) also called machine learning. The work that has been done is this area using soft computing achieves the prediction accuracy for about 12%. One main objective is to design the models with maximum prediction accuracy. The overall objective of this thesis work includes:

i. Thorough survey and understanding on the work that been previously done in this area.
ii. Literature study to identify and investigate the factors that influence the design of AI model to predict next move of mobile host.
iii. Implement realistic mobility model to generate the dataset for the random movement of the mobile host.
iv. Design and implement models using AI to predict the next move of mobile host, taking into consideration the specific requirements.
v. Study of the impact of various parameters of models & training approaches and observe the results of AI model instead of conventional routing.
vi. Evaluate and compare the results.

1.7 Research Methodology

For generating the dataset, the movement of a mobile host is moving for several time intervals using Gaussian Markov Model. A recursive method is used for multiple moves. In this method, the output is inserted as one of the inputs in sub pattern at the extreme right by shifting the entire sub-pattern to left by one time interval to predict the next movement as shown in Table 1.

Sub Pattern	Inputs 1	Inputs 2	Inputs 3	Inputs 4	Desired Output
1	p_1	p_2	p_3	p_4	p_5
2	p_2	p_3	p_4	p_5	p_6
3	p_3	p_4	p_5	p_6	p_7
4	p_4	p_5	p_6	p_7	p_8
5	p_5	p_6	p_7	p_8	p_9
6	p_6	p_7	p_8	p_9	p_{10}

Table 1: Training data set

Movement pattern (p_n) is the history of movement of a mobile host recorded for a period of time T_n, where n is the number of regular time intervals at which the mobile host movements are recorded. The training data set is the set of subpatterns obtained from the movement pattern P_n by partitioning it into n - k subpatterns, where k + 1 is the size of each subpattern (k << n). The subpattern is a training data pair with mobile movements for k time intervals as input and the movement for the next time interval as a desired output. For example, the first training subpattern is p_1, p_2 ... p_k as input and p_{k+1} as the desired output. After generating the dataset, AI models will be designed which will be trained with the history of the movement of the mobile node and then the models will be predicting the next move of the mobile nodes.

1.8 Performance Evaluation

There are a number of measurements for performance evaluation used in literature, we considered the following commonly used ones in the evaluation of the performance of the results for this study.

Prediction accuracy: It is the accuracy for the models will be calculated by number of times correct prediction of mobile node divided by the total number of times the prediction of mobile node in percentage.

Execution time: It include the execution time which gives the output in time (seconds) both for training and testing phase for the prediction model.

The matrices for evaluating the performance of routing protocol will include

i. Average packet delay
ii. Average successful transmissions
iii. Jitter
iv. Energy – Transmission Power.

CHAPTER 2

RELATED WORK

Real-time traffic over ad-hoc is important in two aspects. Firstly, the gradual popularity of ad-hoc networks especially for certain circumstances such as research conference. Secondly, the need to make some particular kinds of network, suitable for real time traffic. Thus, on discussing the related work, we find research in both of these two aspects. Some concentrate on ad-hoc routing protocols to optimize the overall performance while others are more interested in user's data rate and codec's used for compression or decompression. This review mainly focuses on efficient video transmission schemes, power minimization models in MANET and video compression schemes. The authors in [64] introduce multistream coding with multipath transport (MPT) for video traffic over ad-hoc networks. First, a video bit stream is divided into several substreams by video encoder and then packets from different substreams are sent along several different paths. At the receiving end, the procedure is reversed. Moreover, to make multistreaming appropriate for ad-hoc, they propose three different multistreaming techniques, each with different advantages. Morkovian and OPNET model analysis show considerable improvement in video quality [64] but the authors did not use the PCF (Point Coordination Function) mode. PCF mode is necessary for real time applications. The authors [64] progressed step by step and introduced the ideas of multistreaming in a given video data [65] and [66]. Also, the three coding schemes they used have been described in detail in [67], [68] and [69]. An interesting analysis is carried out using H.264 video standard to study video transport over ad-hoc networks in [70]. Their simulation shows that packet size of as small as 300 bytes should be used under unfavorable condition as any increase results in degradation of Peak Signal to Noise Ratio (PSNR). For higher error probabilities, even smaller than 300 bytes is good for significant PSNR. Regarding retransmission attempts, 3 per Message Protocol Data Unit (MPDU) gives the highest achievable PSNR. Results illustrate that greater than three is useless as far as PSNR is concerned. This is useful analysis and good approximation to calculate packet size, error probabilities etc. for a given PSNR [70]. However, they did not present analytical support to their simulation results nor did they exploit the idea of MPT. Moreover, their experiments are sound only for static devices and do not give any guarantee when the devices are mobile.

Among the three schemes described in [64], multiple description motion compensation (MDMC) is unique in the sense that it gives better error resilience on lossy packet networks. The study of [71] gives an analytical model to prove feature of MDMC. Furthermore, the proposed analytical model has also been tested by simulation in the same paper. While the above discussion is related to video standards or ad-hoc networks algorithm to improve the performance, [72] discusses, in general for wireless networks, how to obtain optimal data rate for video traffic. It is well known that video data is sensitive to both delay and delay jitter. In real wireless network traffic, there is a loss of packets due to antenna poor efficiency, interference, fading, and weather conditions etc. resulting in need for retransmission (Automatic Repeat Requests) ARQ or some mechanism to recover the original signal such as Forward Error Correction (FEC). Moreover, buffers play an important role in delay jitter in ARQ. Hence, they also develop a control mechanism for buffers with the objective to minimize delay jitter. In light of previous discussion, it is clear now that use of MPT gives flexibility over data rate. Higher data rate can be attained if routes are independent. The authors in [74] add to it and present a video distortion model. This paper further claims that distortion is due to mainly two factors, namely encoder distortion and packet loss due to congestion. The authors of [74] also use simulation to study the tradeoff between the data rate and congestion. Moreover, the appealing is the comparison results of encoded transmission and decoded transmission which give a quantitative measure of how much distortion is added due to congestion. Another concept of cross-layer design framework is introduced in [73] for real time traffic using H.264 codec in their model. The main idea behind this is that the scheme tries to make maximum use of networks resources such as bandwidth. To avoid congestion, traffic flows and link capacities (on the chosen links) are allocated together. Their simulation shows improvement over traditional schemes in terms of date rate and PSNR. The improvement is multiplied further if we combine MPT with cross-layer design. In [9] the authors are applying Multi Layer Neural Network (MLNN) for mobile user tracking among the Self-Organized Maps (SOM). SOM produces a low-dimensional, discretized representation of the input space of the patterns also called a map. In [1], the author proposed a back propagation algorithm for next-move prediction for the implementation of the selective reservation concept which concludes that the prediction algorithm can be applied in cases of uniform, regular or deterministic movements of the mobile user. The next-move prediction in those cases is reliable, because of the high prediction accuracy achieved. For the random moment of mobile node, there outcome was around 12% accuracy. In [10], the author proposed a prediction-based location management in a mobile network. The approach uses a multilayer neural network to predict the future location of a mobile host based on the history of movement pattern of a mobile host. The MLNN model for single and multiple move prediction is designed for predicting the future location of a mobile host. The performance of the method has been verified for prediction accuracy by considering different movement patterns of a mobile host and learning accuracy of the MNN model. Simulation is also carried out for different movement patterns (i.e. regular, uniform, and random) to predict the future location of a mobile host. The average prediction accuracy was measured and achieved up to 93% accuracy for uniform patterns, 40% to 70% for regular patterns, and 2% to 20% for random movement patterns. The author in [10] is predicting both the direction and the distances at the same time, whereas the author in [1] is only predicting one parameter that is the direction. In [6], the authors propose the use of braided multipath instead of completely disjoint multipath so as to keep the cost of maintaining the multipath low. The costs of such alternate paths are also comparable to the primary path because they tend to be much closer to the primary path. In [7], the authors propose an algorithm which will route data through a path whose nodes have the largest residual energy. In this way, the nodes in the primary path will not deplete their energy resources through continual use of the same route thus achieving longer life. In [8], the authors propose use of a set of sub-optimal paths occasionally to increase the lifetime of the network. These paths are chosen by means of a probability which depends on how low the energy consumption of each path is. The path with the largest residual energy when used to route data in a network may be very energy-expensive too. So, there is a tradeoff between minimizing the total power consumed and the residual energy of the network. The authors propose an algorithm in which the residual energy of the route is relaxed a bit to pick a more energy efficient path [75].

CHAPTER THREE

MANET ROUTING PROTOCOLS

The routing protocols meant for wired networks cannot be used for mobile ad-hoc networks because of the mobility of networks. The vision of mobile ad hoc networking is to support robust and efficient operation in mobile wireless networks by incorporating routing functionality into mobile nodes. Ad-hoc mobile routing protocols can be categorized into three categories, i.e, table driven proactive, on-demand-driven reactive source initiated and the hybrid protocols. We list these protocols and describe them briefly as follows. Table 2 shows some of the popular protocols of each category [78].

i. Table-Driven Routing Protocols or Pro-active Protocols
ii. On-demand Routing Protocols or Reactive Protocols
iii. Hybrid Routing Protocols

Table Driven	On-Demand	Hybrid
Destination Sequenced Distance Vector (DSDV)	Stability Routing Protocol (SSR)	Zone Routing Protocol (ZRP)
Wireless Routing Protocol (WRP)	Dynamic Source Routing (DSR)	Hazy Sighted Link State Routing protocol (HSLS)
Clusterhead Gateway Switch Routing protocol (CGSR)	Temporary Ordered Routing Algorithm (TORA)	Sharp Hybrid Adaptive Routing Protocol (SHARP)
Source Tree Adaptive (STAR)	Ad-hoc on Demand Distance Vector Routing (AODV)	WEAC/VBS-O Protocol
Fisheye State Routing (FSR)	Associativity Based Routing Protocol (ABR)	

Table 2: MANET Routing Protocols

3.1 Table-Driven Routing Protocols

In proactive protocols, nodes continuously search for routing information within a network, so that when a route is needed, the route is already known. Some of the examples of this protocol are shown in Table 2. Next, we describe the working principle of each protocol very briefly to have some review about the characteristics of existing protocols.

3.1.1 Destination Sequenced Distance Vector (DSDV)

This protocol is based on the Bellman-Ford algorithm [74]. It is a distance-vector style protocol for ad-hoc networks. It uses sequence numbers to mark each node and allows far-distant computers / devices to exchange their information along a multi-hop communication [46]. Each node has a table in which the information per routing entry contains the destination address, the number of hops to the destination and the sequence number corresponding to the destination. Tables are periodically updated and broadcast. The detailed analysis of the protocol can be found in [46].

3.1.2 Wireless Routing Protocol (WRP)

This protocol is based on finding paths for a finite number of times. It forces each node to perform consistency checks of predecessor information reported by all its neighbors which ultimately eliminates looping situations and provides faster route convergence when a link failure event occurs. This protocol offers high overhead as far as table entries are concerned because each node has to maintain Distance table, Routing table, Link-cost table & Message retransmission list table. [47] Describes complete analysis of the protocol.

3.1.3 Clusterhead Gateway Switch Routing protocol (CGSR)

This protocol is very similar to DSDV but differs from it in the type of addressing and network organization scheme. In CGSR, mobile nodes are grouped into clusters and each cluster has a cluster head which controls a group of ad-hoc hosts. Moreover, it employs several heuristic routing schemes to be employed. However, frequent cluster head changes can adversely affect routing protocol performance since nodes are busy in cluster head selection rather than packet relaying. This situation is even worse in case of real time traffic as the routing delay keeps on increasing. More details are available in [48].

3.1.4 Source Tree Adaptive Routing (STAR)

Perhaps, this was the first proactive routing protocol that works with link-state information as it does not require periodic routing updates. Moreover, it does not attempt to maintain optimum routes to destinations. Simulation results in [49] show that STAR is an order of magnitude more client than any topology-broadcast protocol. The results also show that STAR is even more efficient than the Dynamic Source Routing protocol which is a very efficient on-demand routing protocol.

3.1.5 Fisheye State Routing (FSR)

This protocol focuses on reducing update overhead (control packets etc.) in large networks by introducing a multi-level fisheye scope. Depending upon the distance to destination, nodes exchange update information. This concept is very useful for the application where less delay is required. This implies that FSR is more desirable for large mobile networks where mobility is high and the bandwidth is low. The simulation results in [50] show that FSR is simple, efficient and scalable routing solution in a mobile ad-hoc environment. Detailed description of the protocol can be found in [50].

3.2 On-Demand Routing Protocols

Demand driven or the source initiated protocols form the second category of ad-hoc mobile routing protocols. These protocols create routes only when desired by source nodes. When a node requires a route to destination, it initiates a route discovery process within the network. This process completes once one route is found or all possible route permutations are examined. Once a route is discovered and established, it is maintained by route maintenance procedure until either a node on the route becomes inaccessible or the route becomes undesired (some other e client route is discovered, for example). Example protocols are given in Table 2. A brief description of the most common on-demand routing protocols is given below.

3.2.1 Associativity Based Routing Protocol (ABR)

ABR, proposed in 1996, is a source-initiated routing protocol that does not need for periodic updates. Each node generates periodic hello messages to show its existence to its neighbors. These hello messages are used to update the association table of each node. Based on the stability of the links, this protocol selects route for transmission. ABR consists of 3 phases, namely Route Discovery, Route Repair/Reconstruction and Route Delete. The study in [51] gives a comprehensive analysis of this protocol.

3.2.2 Signal Stability Routing (SSR)

This protocol is descendant of ABR. It is further divided into two sub protocols based on their respective functionality, i.e. Static Routing Protocol (SRP) and Dynamic Routing Protocol (DRP). SSR selects routes based on signal strength between nodes and a node's location stability instead of just temporal stability (in ABR). DRP is responsible for maintenance of signal stability table and routing table while SRP processes packets by passing the packets up the stack if it is the destination and forwarding the packet if it is not the intended receiver. [52] describes the protocol in detail.

3.2.3 Dynamic Source Routing (DSR)

The DSR protocol [53] is based on the Link-State-Algorithms which mean that each node is capable of saving the best route to a destination. It is based on the concept of source routing. There are 2 major phases of the protocol - route discovery and route maintenance. When a mobile node has a packet to send to some destination, it first checks the cache whether there is an entry for that destination. If yes, then it uses that path to transmit the packet. It also attaches its source address on the packet. If there is no entry in the cache or the entry is expired (due to being unused for a long time), the node broadcasts a route request packet to all its neighbors requesting for a path to the destination. Until the route is discovered, the node will be waiting and during this time it can do other things like sending other packets or forwarding packets to other destinations. When the route request packet arrives to any other nodes, they check whether they know the destination asked (may be from neighbor or from their caches). If they have route information, they send back a route reply packet to the destination. Otherwise they broadcast the same route request packet. Once the route is discovered, the sender will send the required packets to the destination using the discovered route and insert an entry in the cache for future use. The node also keeps age information on the entry to recognize whether the route data is fresh or not. When any intermediate node receives a data packet, it first sees whether the packet is sent to itself or not. If it is the destination, it receives it otherwise it forwards the packet using the path attached on the packet. Route maintenance is done by the use of route error packets and acknowledgments. As ad-hoc networks are very promiscuous, any link might fail at any time. So the route maintenance process monitors the route and notifies the nodes any failure along its path and the nodes change the entries of their route cache accordingly.

3.2.4 Temporary Ordered Routing Algorithm (TORA)

TORA (Temporary Ordered Routing Algorithm) is an adaptive and e client routing algorithm primarily designed for Mobile Ad-hoc Networks (MANETs). It comes under the category of on-demand MANET routing protocol (with source-initiated feature). TORA is a fairly complicated protocol but what makes it unique and prominent is its main feature of propagation of control messages only around the point of failure when a link failure occurs. Other protocols, on the contrary, need to re-initiate a route discovery when a link fails. TORA would be able to patch itself up around the point of failure. This feature allows TORA to scale up to larger networks but has higher overhead for smaller networks. Moreover, in its enhanced version, it stores the time value since a link failure. The protocol algorithm has three distinct phases, namely route creation, route maintenance and route erasure. In route creation, routes are created mostly in a reactive mode. Initially, all nodes are disconnected. The protocol then forms a DAG (Directed Acyclic Graph). The criterion of adding node is based on a metric called height. The node j is added wit the node i, which is already member of the DAG if $h_i > h_j$. The metric height consist five arguments all of which define the height of the node. The source sends QRY (query) packet indicating the destination node. The QRY packet propagates until it reaches a node whose neighbor is the specified destination, which then transmits a UPD (update) packet. All is done locally, i.e. the nodes know only their neighbors and not all members of the network. In route maintenance, route is maintained only for nodes with non-null height. On link failure, if a node is not connected to any node with a height smaller than its own, all of its links are reversed based on link reversal algorithm. This is how routes are adapted according to topological changes. This feature adds extra overhead even if that path is not required for data transmission. For this reason, TORA is also considered member of the Table-Driven MANET protocols. A route is erased on the reception of the CLR (clear) packet from a source in route erasure phase. A node, on receiving the CLR packet, sets its own height and heights of all its neighbors to NULL and broadcasts the CLR packet. This way, route erasure is performed [54].

3.2.5 Ad-hoc on Demand Distance Vector Routing (AODV)

This routing algorithm [29] is entirely based on the DSDV algorithm with the improvement in minimization the number of required broadcasts by creating routes on-demand (unlike DSDV). Broadcast is used for route requests and a path discovery is initiated when a route to a destination does not exist (like DSR). Similarly link failure is handled much similar to that we discussed in DSR previously. The advantage of using AODV is e client use of available bandwidth as it minimizes the network load for control and data traffic. Moreover, the algorithm is scalable, handles changes in topology, and ensures loop free routing. Entries of a routing table include Destination IP Address, Prefix Size, Destination Sequence Number, Next Hop IP Address, Lifetime, Hop Count, Network Interface and other state and routing flags (valid, invalid).

3.3 Hybrid Approach

As the name suggests, hybrid has features of both table driven and on-demand routing schemes. These protocols exploit the merits of on-demand and proactive routing protocols, thus achieving overall better performance since they use both reactive and proactive schemes. On the other hand, the path to a destination may be suboptimal on account of hierarchical routing. Furthermore, memory requirement is greater because each node has higher level topological information. We give brief overview of the hybrid protocols tabulated in Table 2 in the subsequent sections.

3.3.1 Zone Routing Protocol (ZRP)

ZRP [56] is a typical example of hybrid type ad-hoc routing protocol. The network is divided into different zones by ZRP which are named as nodes local neighborhood. The basic idea behind introducing ZRP is the proactive advertisement of each node of its link state over a fixed number of hops which is called the zone radius. These advertisements (within the zone radius) keep on providing updated routing conditions to each node, i.e. the collection of all nodes and links that are reachable within the zone radius. Like Border Mobile Terminal (BMT) nodes in the WEAC protocol, there are peripheral nodes at the boundary of routing zone that play an important role in the reactive zone-based route discovery.

3.3.2 Hazy Sighted Link State routing protocol (HSLS)

Basically, the Hazy-Sighted Link State Routing Protocol (HSLS) is a Link-state routing protocol. It is an algorithm for computers communicating by digital radio in a mesh network to find each other, and communicate over a reasonably efficient path. The analysis in [57] shows that the HSLS protocol gives one the best performance behavior on routing overhead.

3.3.3 Sharp Hybrid Adaptive Routing Protocol (SHARP)

SHARP has also hybrid approach for ad-hoc routing. Its beauty lies in the fact that it automatically finds a balance between proactive and reactive routing by adjusting the degree to which route information is propagated proactively versus the degree to which it needs to be discovered reactively. Moreover, the proactive routing component is less expensive than that of the ZRP protocol [58]. Analysis in [58] shows that this protocol supports applications having different demands to control the performance of the routing layer. It can be used to minimize packet overhead, limit packet loss rate, and control delay jitter.

3.3.4 WEAC / VBS-O Routing Protocol

Warning energy aware clusterhead (WEAC) / Virtual base station on demand (VBS-O) protocol establishes a dynamic wireless mobile infrastructure in MANET. A mobile node is elected from a set of nominees to act as a temporary base station for a period of time within its zone. There are basically four types of nodes in this protocol:

• **Clusterhead (or VBS)** - the leader of the cluster. Collects complete information on all other cluster heads and their lists of MTs and broadcasts this info in its periodic hello message and identification used is myCH==0 (variable name) for ClusterHead.

• **Zone MT** - An MT (mobile terminal) supervised by a clusterhead. Its myCH is always greater than zero. The value for the variable myCH=ID number of its clusterhead. It accumulates information about the network from their neighbors between hello messages and broadcasts to its neighbors, announcing its ID number with its periodic hello messages.

• **Free MT** - An MT that is not associated with a cluster. It comes into existence when an MT's power level goes below Threshold Th3 then its myCH is assigned -1.

• **Gateway or Border Mobile Terminal** - An MT that lies between more than one clusterhead or Free MT. In other words, it joins two clusters or one cluster with a node that is not part of that cluster. BMT can be zone MT, free MT or clusterhead. Accordingly, the value of myCH can have any value depending upon its type.

As this protocol is power-aware, its working principle spins around the node power. A node's power can be in one of three states as shown in Fig. 1 by Threshold Th1, Threshold Th2 and Threshold Th3. If the power of a node is greater than Threshold Th3 (75% of the maximum power in our simulation design) then it can act as a CH (ClusterHead), i.e. it can respond to MergeRequest message. If it is already CH, it can retain its status.

If the power of a node lies in the interval Th2 < node's power < Th3, where Th2 is 50% of the maximum power, then it cannot respond to MergeRequest message positively. However, if it is already CH, it can remain CH. If the power of a node lies in the interval Th1 < node's power < Th2, where Th1 is 25% of the maximum power, then it cannot respond to MergeRequest message positively. If it is already CH, it sends a Warning Message to all of its zone MTs indicating that they should start searching for new CH with power greater than T h1. However, it can remain CH. If the power of a node is less than Th3, then it immediately sends IamNoLongerYourCH to its zone MTs and sets it myCH == -1 (becomes free MT). All zone MTs also set their myCH to -1.

Fig. 2 shows merge and accept processes for the WEAC protocol. If a free MT or a node whose CH has low power (< Th3) finds a node whose power is greater than Th1 and it is not a zone MT then the first node sends a MergeRequest message to the latter node. Upon receiving, the second node sends a MergeAccept message back to the sender node, sets its myCH to zero (if it was not zero before) and increments its number of zone MT variable.

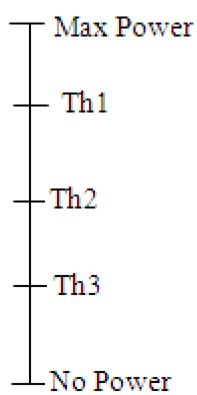


Fig. 1: Different Power Levels of WEAC Protocol

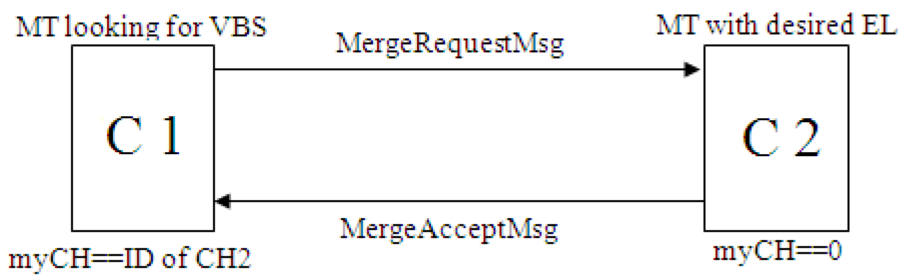


Fig. 2: WEAC MergeAccept Scenario

The WEAC protocol is hybrid type (on-demand/table driven) with Cluster formation (CH, ZMT, BMT, free-MT) and a node can be in one of the following states:

i. Idle
ii. Medium access (CSMA/CA)
iii. Transmit (data / control packet)
iv. Receive (data / control packet)
v. Power check
vi. Mobility

The authors of [59] studied the characteristics and performance of both the WEAC and the VBS-O protocols by means of simulation. In addition, clusterhead controlled token is used to assign the channel among contending Mobile Terminals (MTs) in each cluster. They show that both the WEAC and the VBS-O protocols scale well to large networks of mobile stations, and outperform other power aware routing protocols (e.g., Virtual Base Station, Power Aware-VBS) in terms of stability, load balancing and energy saving in a network.

CHAPTER FOUR

MOBILITY MODELS

The application of mobility model is of great importance because it describes the movement pattern of mobile users by explaining how their location, velocity and acceleration change with respect to time. These mobility models have concrete reasoning or benchmarks to evaluate the end products performance. So it is very much necessary to use the mobility models to follow the movement pattern of targeted real life applications in a realistic way. Every mobility model has different characteristics. So anyone can be used based on their research requirements. Such models can provide performance parameters for simple cases through mathematical calculations. In contrast, simulation models consider more detailed and realistic mobility scenarios. Such models can derive valuable solutions for more complex cases. The details of various mobility models that are used in the field are described below.

4.1 Random Walk Mobility Model - RWMM

Naturally many entities move in exceedingly erratic ways, the random walk mobility model was developed to mimic this erratic movement. In this mobility model, mobile node moves from its current location to a new location by randomly choosing both the direction and speed. The new speed and direction are both chosen from ranges defined in advance [speedmin, speedmax] and [0, 2π] respectively. The movement can be calculated in two ways; either with a constant time interval *t* or with a constant distance traveled *d*. If mobile node approaches the boundary, it bounces back with an angle determined by the next upcoming direction. The 2-D random walk mobility model is one of the derivatives and is inspired by the Earth's surface. Fig. 3 shows the movement observed from this 2-D model. The MN starts its movement in the center of the 300mx600m simulation area (150, 300). At each point, the mobile node randomly chooses a direction between 0 and 2π and a speed between 0 and 10 m/s. For the random walk mobility model, the mobile node movement can be modeled in two ways; either with distance or with time. In Fig. 3, the network is designed so that the MN will calculate the new distance and speed after 60 seconds that is after constant time period whereas in Fig. 4, the mobile node travels for a total of 10 steps before changing its direction and speed. But it is a memory-less mobility pattern [1][10][29].

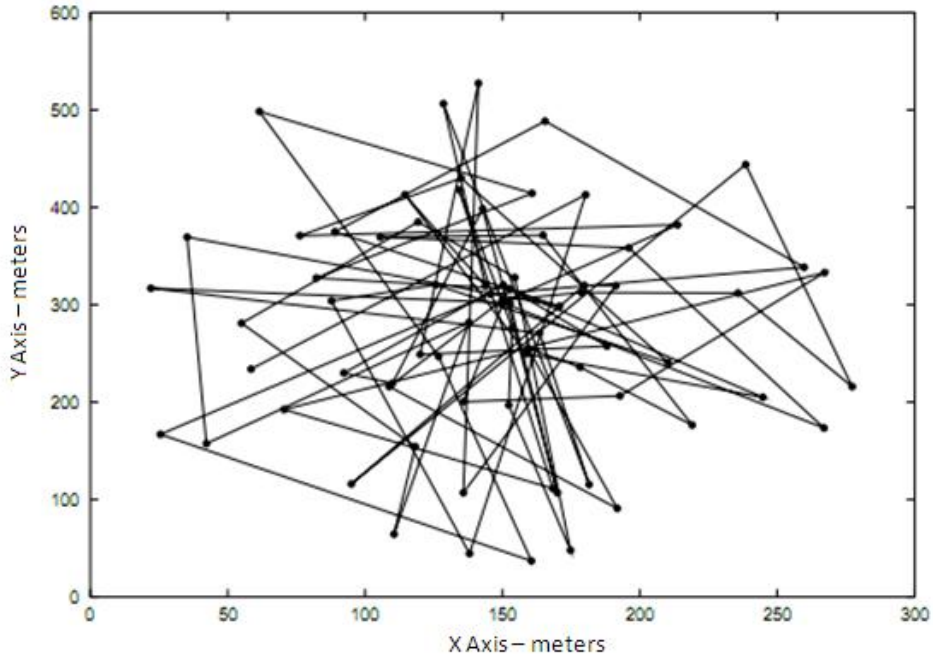


Fig. 3: Movement pattern of mobile node using 2-D Random Walk Mobility Model (time) (300x600).

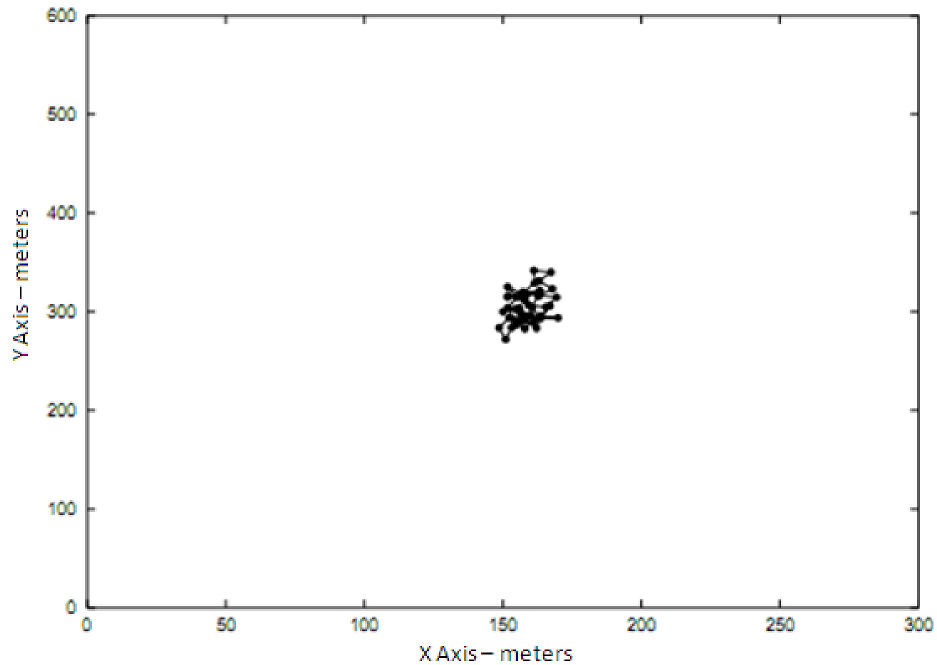


Fig.4: Movement pattern of mobile node using the 2-D Random Walk Mobility Model

(distance) (300x600).

4.2 Random Waypoint Mobility Model

The random waypoint model (RWMM) was proposed by Johnson and Maltz in [5]. It is a model that includes pause times between changes in destination and speed. Firstly the mobile node chooses a random location and considers it as it destination and then it moves towards its destination with constant velocity which is uniformly distributed between [minvelocity, maxvelocity]. After arriving at the destination, the MN pauses for a specific time before choosing another random destination. The pause time can have the value zero '0' which means that it will continue its movement without any pause. Speed is uniformly distributed between [minvelocity, maxvelocity]. The MT then travels toward the newly chosen destination at the selected speed. Upon arrival to the simulation boundary, the MN pauses for a specified time period before starting the process again with an angle determined by the incoming direction. [16][4][18].

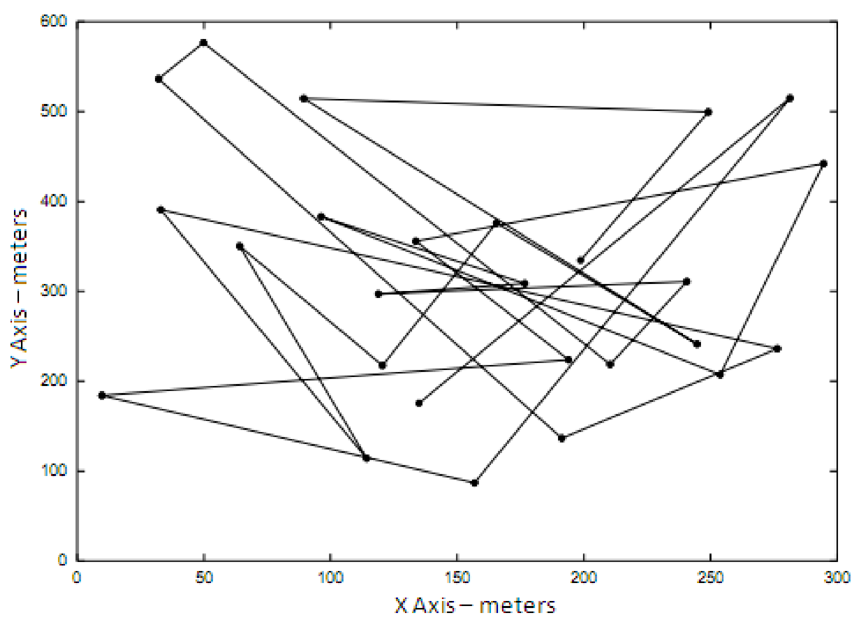


Fig. 5: Movement pattern of mobile node using the random waypoint mobility model.

4.3 Random Direction Mobility Model

The characteristics of random waypoint mobility model was that it produces density waves in the average number of neighbors, so to triumph over this, it gave the reason to create Random Direction. A density wave appears to be forming clusters of nodes in particular part of the simulation grid. For the case of the random waypoint mobility model, the probability for choosing its next destination to the center in very high which result in convergence of mobile node movement in one area. So the random direction mobility model is developed keeping that thing in mind. In this model, MNs choose a random direction in which to travel similar to the random walk mobility model. Mobile node then travels to the border of the simulation area in that direction. Once the simulation boundary is reached, the MN pauses for a specified time, chooses another angular direction (between 0 and 180 degrees) and continues the process. So once the MN starts its movement, it will move in the same direction and will not change the angle until it reaches the border grid. Fig. 6 shows an example trajectory of an MN, which starts from the center of the simulation area (150, 300) using the random direction mobility model. The dots in the figure show when the mobile node has reached a border, pauses, and then chooses a new direction. Since the mobile node travels to, and usually pause at the border of the simulation area, the average hop count for data packets using the random direction mobility model will be much higher than the average hop count of most other mobility models. In addition, network partitions will be more as compared to other mobility models [24].

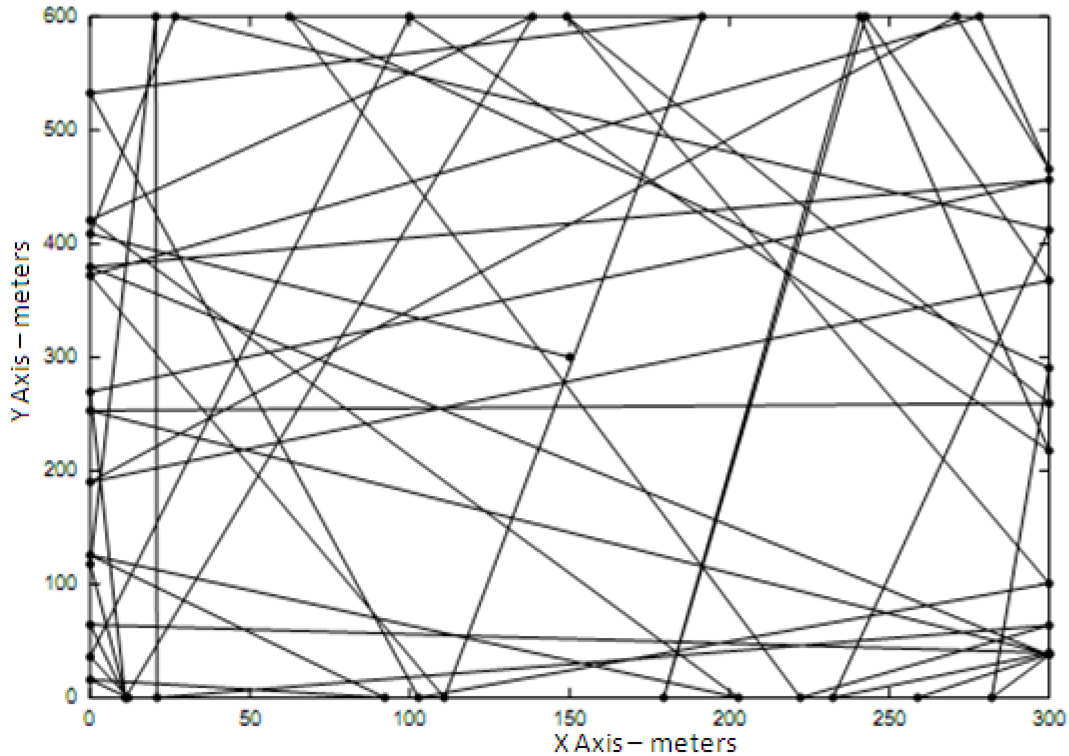


Fig. 6: Movement pattern of mobile node using the Random Direction Mobility Model.

4.4 Gauss-Markov

From [9], the Gauss-Markov Mobility Model is planned to achieve randomness via one tuning parameter. Initially each mobile node is assigned a current speed and direction. At fixed intervals of time, movement occurs by updating the speed and direction of each MN. Specifically, the value of speed and direction at the nth instance is calculated based upon the value of speed and direction at the (n-1)st instance and a random variable using the following equations:

$$s_{n=}\alpha s_{n-1} + (1-\alpha)\bar{s} + \sqrt{(1-\alpha^2)}s_{x_{n-1}}$$

$$d_{n=}\alpha d_{n-1} + (1-\alpha)\bar{d} + \sqrt{(1-\alpha^2)}d_{x_{n-1}}$$

where s_n and d_n are the new speed and direction of the MN at time interval n; where alpha value lies within the range $0 \le \alpha \le 1$. It is the tuning parameter used to vary the randomness; s and d are constants representing the mean value of speed and direction as $n \to \infty$; $s_{x_{n-1}}$ and $d_{x_{n-1}}$ are random variables from a Gaussian distribution and are chosen with mean equal to zero and standard deviation equal to one. For α = 0 the equation yields totally random values, equivalent to Brownian motion. For α = 1 the equation yields fixed values, equivalent to linear motion. The value of α can be adjusted between these two extremities to obtain different levels of random movement. At every time interval the next location of the mobile node is calculated based on the current location, speed, and direction of movement. Specifically, at time interval n, an MN's position is given by the equations:

$$x_n = x_{n-1+}s_{n-1}cosd_{n-1}$$

$$y_n = y_{n-1+}s_{n-1}sind_{n-1}$$

where (x_n , y_n) and (x_{n-1} , y_{n-1}) are the x and y coordinates of the MN's position at the nth and (n-1)th time intervals, respectively, and s_{n-1} and d_{n-1} are the speed and direction of the MN, respectively, at the (n-1)th time interval.

To ensure that mobile node does not remain near an edge of the grid for a long period of time, the MNs are forced away from an edge when they move within a certain distance of the edge. This is done by modifying the mean direction variable d in the above direction equation as shown in Fig. 7. For example, when mobile node is near the right edge of the simulation n grid, the value *d* is changed by 180 degrees. Thus, the MN's new direction is away from the right edge of the simulation grid. So in short, for the Gauss-Markov model, the velocity of a mobile node at any time slot is a function of its previous velocity. So, Gauss-Markov Model is a mobility model with chronological dependency. The degree of dependency is determined by the memory level parameter α.

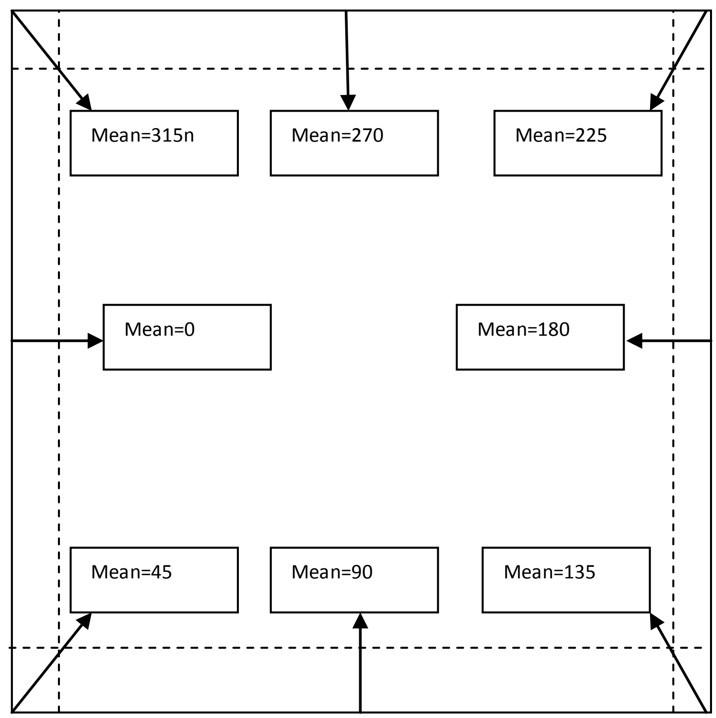


Fig. 7: Edge approximation for changing angle near edges

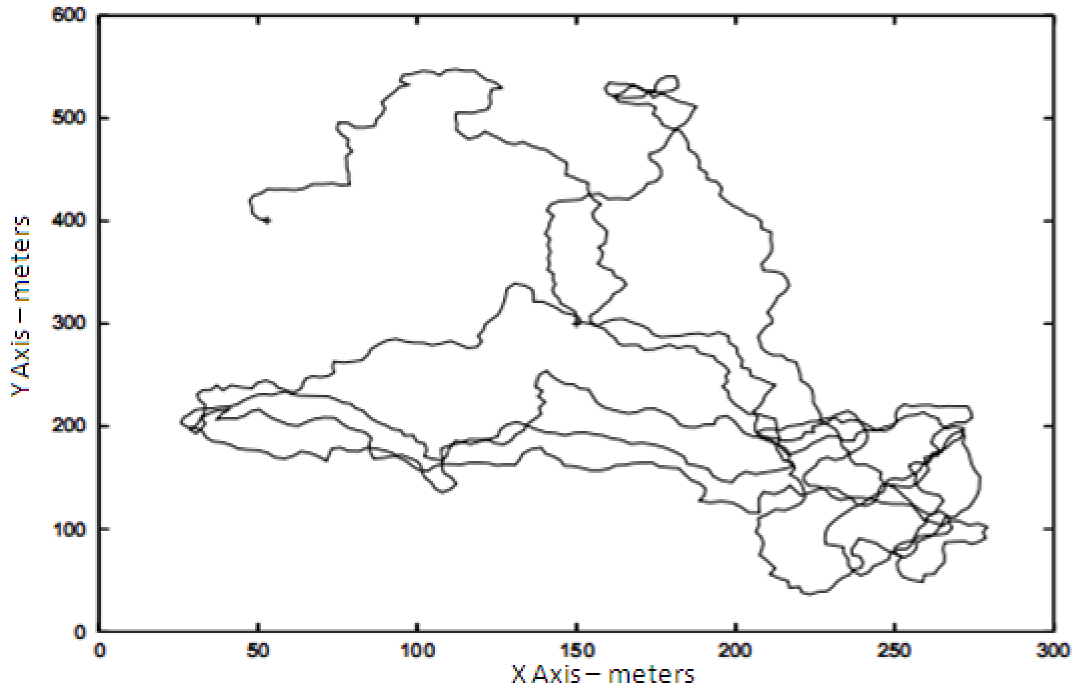


Fig. 8: Expected movement pattern of Gaussian Markov model

4.5 A Boundless Simulation Area

In the Boundless Simulation Area Mobility Model, a correlation exists between the previous and current direction and velocity of mobile nodes. A velocity vector v = (v, θ) is used to describe a node's velocity v as well as its direction θ; the mobile nodes position is represented by x and y. Both the velocity vector and the position are updated after Δt time according to the following formulas:

$$v\,(t+\Delta t) = \min[\max(v(t) + \Delta v, 0)\,, V_{\max}];$$

$$\theta(t\Delta t) = \theta\,(t) + \Delta\theta;$$

$$x\,(t+\Delta t) = x(t) + v(t) * \cos\theta\,(t);$$

$$y\,(t+\Delta t) = y(t) + v(t) * \sin\theta(t);$$

Where V_{max} is the maximum velocity defined in the simulation, Δv is the change in velocity which is uniformly distributed between [- $\alpha * \Delta t$, $\alpha * \Delta t$] where α is the angular change if the direction in which mobile node is moving [12]. The movement pattern is shown in Fig. 9.

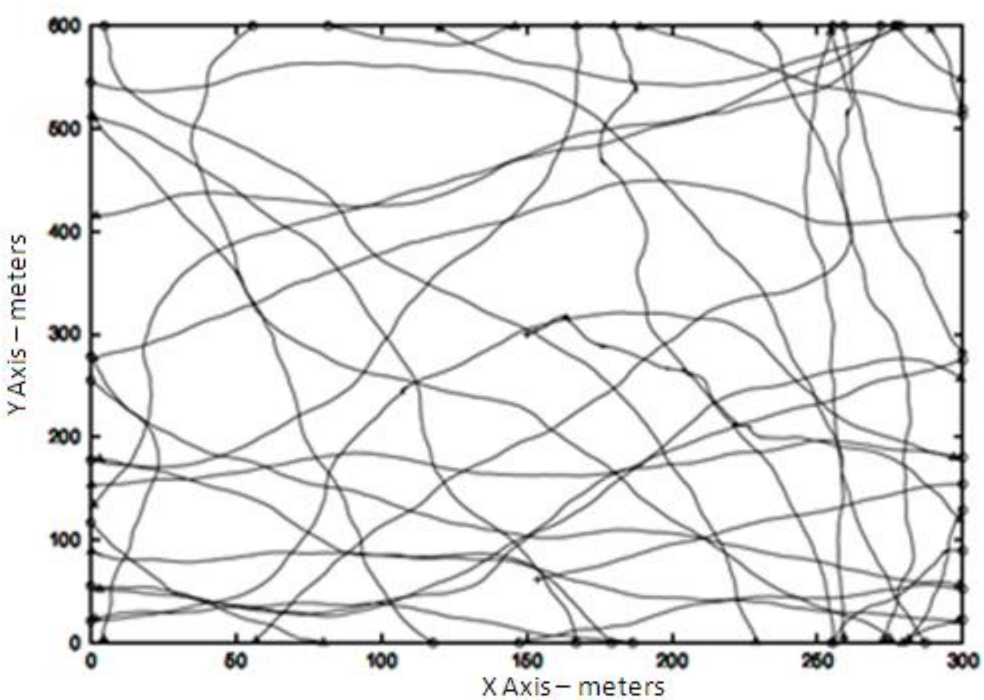


Fig. 9: Movement pattern of mobile node using Boundless Simulation Area Mobility Mode.

CHAPTER FIVE

NEURAL NETWORKS

Artificial Neural Network (ANN) or commonly called Neural Network (NN) is an information processing prototype that is inspired by the way biological nervous systems, such as human brains. Many similarities are found between the working of brain and ANN. Human learns by example and experience. Neural Networks (NN) is composed of a fair number of highly interconnected processing elements called neurons that work in unison to solve specific problems [40]. It carries out certain computations many times faster than the today's computer. NN assigns weights to the neurons that are learnt from data using some known algorithms. Network topology is defined by the initial weights, activation functions, number of neurons in hidden layers, and network connections. The performance of ANN depends on the network topology which in turn depends on the problem to be solved. Similarly, an ANN is also configured for a specific application, such as pattern recognition or data classification, through a learning process. The more the samples size, the better the ANN is trained or learned. Moreover, learning in brain involves adjustments to the synaptic connections that exist between the neurons present inside the brain [20, 21, 22]. This is true for an ANN as well where weights, that are scalar values associated to neurons, are adjusted as the training proceeds. Hence, ANN is an efficient tool that tends to imitate processing of brain and has been in use for classification, function approximation, regression, and estimation. It is proved to be better in performance with respect to many attributes when compared with other classifiers [23]. However, it was established before the advent of computers. The first artificial neuron was invented by a neurophysiologist, Warren McCulloh, and the logician, Walter Pits, in 1943. The recourses and technology available at that time does not allow the field of neural networks to prosper much. But as the inexpensive computers and higher technology came into existence, it boosted up the advances in neural networks. Although, just after the very start, the neural networks field survived a period of disrepute and frustration for some years, the reemergence of interest and progress later took over. In the late 1970s and early 1980s, the field of neural network started enjoying a resurgence of interest and a corresponding increase in funding. This caused many developments in the neural network field along with its application in several new domains. Neural networks have emerged as an important tool for prediction. The recent times, vast research activities in neural network have established that they are a promising alternative to various conventional methods. The advantage of neural networks is as follows. Input data or signals are fed into the input layer and propagate through the network layers in forward direction. The neurons in the hidden layer find relationships in the input data and take out patterns that can give meaningful results at output layer. These neurons have an activation function and are connected to many other neurons in the neighboring layers.

Each neuron contributes to the final output depending on the activation and the connections. We can train the network by comparing the actual and desired output. If the output of the network is different than what we expect, we can make adjustments in the weights which control activation and the connections. When the error is within the acceptable range, training is stopped, weights are locked, and the network is ready to use. The main characteristics of ANN are firstly, neural networks are data driven self-adaptive methods in that they can adjust themselves to the data without any explicit specification of functional or distributional form for the underlying model. Secondly, they are universal functional approximations, in that neural networks can approximate any function with arbitrary accuracy. Thirdly, neural networks are nonlinear models, which make them flexible in modeling real world complex relationships. Finally, neural networks are able to estimate the posterior probabilities, which provide the basis for establishing classification rule and performing statistical analysis [24].

5.1 Applications of Neural Networks

There are many applications in which neural networks are successfully adopted. Some of the interesting applications of neural networks are presented at [25]. A few examples of ANN application are defined below:

• **Stock price prediction in stock markets**: Neural networks are being used by many technical analysts to make predictions about stock prices based upon a large number of factors such as past performance of other stocks and various economic indicators [26, 27].

• **Medical diagnosis**: A variety of health-related indices (e.g., a combination of heart rate, levels of various substances in the blood, respiration rate etc.) can be monitored. The onset of a particular medical condition could be associated with a very complex combination of changes on a subset of the variables being monitored. Neural networks have been used to recognize this predictive pattern so that the appropriate treatment can be prescribed [28, 29].

• **Computer vision**: Neural network is intensely used in many of the computer vision algorithms such as those in [30, 31].

• **Optical character recognition - OCR**: Neural network is used in optical character recognition for different languages [32].

• **Engine management**: Neural networks are used to analyze the input of sensors from an engine. The neural network controls the various parameters within which the engine functions, in order to achieve a particular goal, such as minimizing fuel consumption [33].

5.2 Advantages of NN

There are a number of benefits of using ANN over other classifiers. Some of them are discussed as follows:

• **Sequential process of information**: One advantage of neural networks techniques that makes them more attractive for use in real time systems is their ability to process information sequentially i.e. inputs can be processed as they arrive. Comparatively, in statistical techniques however, no decision about the classification of any pattern can be made until all the patterns are available.

• **Adaptive learning**: Neural networks posses the ability to learn how to perform tasks based on the data given for training.

•**Self organization**: A Neural network creates its own organization or representation of information it receives in the learning time.

• **No strict assumptions**: Neural networks provide an analytical alternative to conventional techniques which are often limited by strict assumptions of normality, linearity, variable independence etc.

• **Performance**: Neural networks perform very well on difficult non-linear domains where it becomes more and more difficult to use other approaches such as decision trees. They also perform better in noisy domains.

Along with these advantages, there are some limitations and issues that need to be considered when implementing neural networks as described below:

5.3 Limitations

Some of the main limitations of neural networks identified in the literature are described as follows [34, 35]:

i. Slow learning process as compared with statistical classifiers like decision trees.
ii. No explicit knowledge representation in the form of rules or some other easily interpretable forms. The model is implicit and hidden in the network structure and optimized weights, between the nodes. Hence, the model tends to be a black box or input/output table without analytical basis.

iii. Although neural networks are parameter-less and data-driven classifiers, a wise selection of the network structures has intense effects on some important issues in neural networks namely over-fitting and under-fitting.

These advantages and the limitations of neural networks will be taken into consideration when designing the neural network model for prediction.

5.4 Network Generalization

Generalization is one of the important issues which need to be considered in using neural networks. Generalization means how the network will make predictions for cases that are not in the training set. It is a critical issue in developing neural networks. If the network is not trained well and the accuracy in testing on training data as well as evaluation data is not promising then the network is said to be undertrained or underfitted. On the other hand, if the network has learned the training dataset and gives a fairly good accuracy when testing on the training data whereas an unaccepted result when testing on a novel dataset, in this case the network is said to be over-trained or over-fitted. Both under-fitting and over-fitting are against generalization of the neural network and need to be necessarily resolved. There are many strategies to avoid over and under- fitting. We discuss some of them from the literature below:

• **Using a large amount of training data**: A rule of thumb is to use 30 times as many training cases as there are weights in the network. This reduces much of the over-fitting. For simpler (noise free) data five times as many training cases as weights may be sufficient [35].

• **Jittering**: Deliberately adding noise to the input samples during training [36].

• **Early Stopping**: Discontinue training when it seems that the network is being over trained. This is carried out with the help of validation vectors [37].

• **Weight decay**: Adding a penalty term to the error function. Using the weight decay excessively large weights, which can cause the output function to be too rough, possibly with near discontinuities, are avoided in the network.

5.5 Learning Strategies

The network needs to be trained before it is used to produce some outputs. Learning corresponds to the adjustments of weight matrices in the network. There are three types of learning strategies used in neural networks namely supervised, unsupervised and reinforcement learning. Supervised and unsupervised learning schemes will be used which are most common in the literature for ability estimation in traditional test. Supervised learning involves learning a function from examples of its inputs and outputs. However, unsupervised learning involves learning the pattern in the input when no specific output values are supplied.

5.6 Prediction Using Neural Networks

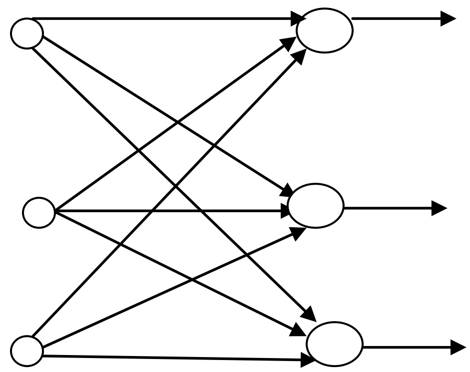
Prediction problems may be divided into two main categories namely classification and regression. For classification, the aim is to find out which number of classes a given input dataset belongs. Examples include cancer detection (e.g. does the patient has tumor or no tumor), signature recognition. On the other hand, the objective of regression is to predict the value of a continuous variable (e.g. tomorrow's stock price, fuel consumption of a car)

5.7 Modeling of Traditional Tests Using Neural Networks

Once training is complete, the neural network can be used to predict the desired output for a given input pattern. There are several flavors of neural network models. In this section, we focus on three famous models: Multi-Layer Perceptron (MLP), Principal Component Analysis (PCA) and Radial Basis Function (RBF). The neural networks used here are briefly described below.

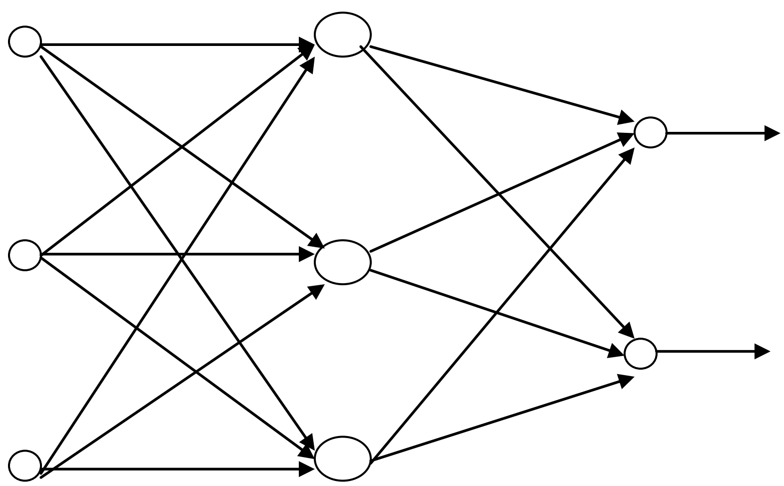
5.7.1 Multi-Layer Perceptron (MLP)

The multilayer perceptron (MLP) is a hierarchical formation of several perceptrons, and is designed to overcome the shortcomings of single-layer networks. In a single layered NN the neurons are organized in the form of layers. In the simplest form of a layered network, we have an input layer of source nodes that projects onto an output layer of neurons but not vice versa. In other words, this network is strictly a feed forward or acyclic type. The single layer refers to the output layer of computation nodes (neurons) as shown in Fig. 10 shows the working of single layered NN. MLP is a feed-forward multi-layer neural network that consists of a collection of neurons or Processing Elements (PEs) organized in layers such that each PE receives inputs only from the PEs in the immediately preceding layer. The weights are tuned iteratively using back-propagation training algorithm to minimize the mean squared error between predicted output and desired output. Fig. 11 shows a representation of MLP. In this example, there are three input nodes, one hidden node and one output in the input layer, hidden layer and output layer respectively. All the nodes in a layer are connected to all the nodes in the next layer.



Input Layer Output layer

Fig. 10: Single layer feed forward network



Input Layer Hidden layer Output layer

Fig. 11: Typical multi-layer perceptron with 3 input nodes, 1 hidden node and 2 output nodes.

5.7.2 Principal Component Analysis (PCA)

PCA has been immensely used in the fields of pattern recognition, statistics as well as signal processing. The basic idea of PCA is to obtain a new set of variables, the Principal Components (PCs), which are uncorrelated and ordered from a dataset consisting of large number of interrelated variables and thus reducing the dimensionality of the input space. In the experiments, a hybrid model is used which consists of two stages. As a first step, it uses PCA with unsupervised learning to reduce the dimensions of the input dataset whereas in the second step, it uses MLP with supervised learning to map the resulting reduced input dataset to the desired output. Four components are used for pre-processing in MLP.

5.7.3 Radial Basis Function (RBF):

RBFs are influential practices for interpolation in multidimensional space. An RBF is a function which has built into it a distance criterion with respect to a center. These types of functions are used very efficiently for interpolation and smoothing of data. In our experiments, Radial Basis Functions have been adopted in neural networks where they are used as a replacement for the sigmoidal transfer function. The neural network has now 2 layers: the hidden layer with the RBF non-linearity and a linear output layer. The best choice for the non-linearity is the Gaussian. RBF networks have the advantage of not being locked into local minima as do the feed-forward networks [38, 39]. Learning of RBF consists of two stages. In the first stage, RBF uses unsupervised learning to perform a non-linear mapping from an input space into a higher dimensional space in which the pattern become linearly separable. In the second stage, supervised learning using a simple MLP is performed.

CHAPTER SIX

EXTREME LEARNING MACHINE (ELM)

The learning speed of feed forward neural networks is in general far slower than required and it has been a major bottleneck in their applications for past decades. Two key reasons behind may be:

i. The slow gradient-based learning algorithms are extensively used to train neural networks
ii. All the parameters of the networks are tuned iteratively by using such learning algorithms.

Unlike these conventional implementations, extreme learning machine (ELM) for single-hidden layer feedforward neural networks (SLFNs) randomly chooses hidden nodes and analytically determines the output weights of SLFNs. Theoretically, this algorithm provides good performance at extremely fast learning speed. The results are based on different artificial and real standard function approximation. Classification problems that include large difficult applications show that the new algorithm can produce good performance and can learn thousands of times faster than conventional learning algorithms for feedforward neural networks. Unlike traditional learning methods, ELM requires less human interventions and can run thousands times faster than conventional methods. ELM automatically determines all the network parameters analytically, which avoids trivial human intervention and makes it efficient in online and real-time applications [41] [42].

Compared to popular Backpropagation (BP) Algorithm and Support Vector Machine (SVM), ELM has several salient features [43]:

Ease of use. No parameters are needed to be tuned manually except predefined network architecture. Not much time is required to tune and train learning machines.

Faster learning speed: The training can be done in milliseconds, seconds, and depending on applications. Such fast pace learning speed might not be easily achieved using conventional learning method.

Higher generalization performance: It could obtain better generalization performance than Back Propagation. It can achieve generalization performance similar to or better than SVM.

Suitable for almost all nonlinear activation functions & fully complex activation functions. Some continuous functions can be used as activation functions. Similarly, fully complex functions can also be used as activation functions in ELM.

In a seminal paper Baum implied that in a Single Layer Feedforward Network (SLFN) one may fix the connections on one level (i.e., weights between input neurons and hidden neurons) and only adjust the connections on the other level (i.e., weights between hidden neurons and output neurons) and there is no gain achieved by an algorithm able to adjust the weights on both levels simultaneously.[44] Inspired by this work, Huang *et al.* have proposed a new learning algorithm referred to as Extreme Learning Machine (ELM) [45]. ELM randomly chooses and fixes the weights between input neurons and hidden neurons based on some continuous probability density function, and then analytically determines the weights between hidden neurons and output neurons of the SLFN.

6.1 Approximation problem of SLFNs

For N samples $\{(x_k, t_k)\}$ N_k=1, where $x_k = [x_{k1}, x_{k2}, \ldots, x_{kn}]\ T$ and $\mathbf{t}_k = [t_{k1}, t_{k2}, \ldots, t_{km}]\ T$,

A standard SLFN with ˜N hidden neurons and activation function $g(x)$ is mathematically

modeled as

$$\sum_{i=1}^{N} \beta_i g(w_i * x_k + b_i) = o_k, \qquad k = 1,2, \ldots, N, \qquad (1)$$

Where $\mathbf{w}_i = [w_{i1}, w_{i2}, \ldots, w_{in}]^{\ T}$ is representing the weight vector that connects the i^{th} hidden neuron and the input neurons, $\beta_i = [\beta_{i1}, \beta_{i2}, \ldots, \beta_{i\ m}]T$ is the weight vector connecting the i^{th} hidden neuron and the output neurons, $\mathbf{o}_k = [o_{k1}, o_{k2}, \ldots, o_{k\ m}]\ T$ is the output vector of the SLFN, and b_i is the threshold of the i^{th} hidden neuron. $\mathbf{w}_i$, $\mathbf{x}_k$ denotes the inner product of $\mathbf{w}_i$ and $\mathbf{x}_k$. And these N equations can be written compactly as:

$$H\beta = 0 \qquad (2)$$

Where

$$H=\begin{bmatrix} g(w_{1.}x_N + b_1) & \cdots & g(w_{N.}x_1 + b_N) \\ \vdots & \ddots & \vdots \\ g(w_{1.}x_N + b_1) & \cdots & g(w_{N.}x_N + b_N) \end{bmatrix}_{N*N} \qquad (3)$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_N^T \end{bmatrix}_{N*m} \qquad (4)$$

$$O = \begin{bmatrix} O_1^T \\ \vdots \\ O_N^T \end{bmatrix}_{N*m} \qquad (5)$$

Where H is called the hidden layer output matrix

6.2 ELM learning algorithm

In the case of learning an arbitrary function with zero training error, Baum had presented several constructions of SLFNs with sufficient hidden neurons [44]. However, in practice, the number of hidden neurons required to achieve a proper generalization performance on novel patterns is much less. And the resulting training error might not approach to zero but can be minimized by solving the following problem:

$$\underset{w_i,b_i,\beta}{min} \|H(w_1,\dots,w_N,b_1,\dots,b_N)\beta - T\|^2, \quad (6)$$

Where

$$T = \begin{bmatrix} T_1^T \\ \vdots \\ T_N^T \end{bmatrix}_{N*m} \qquad (7)$$

ELM randomly assigns and fixes the input weights **w**i and biases bi based on some continuous probability distribution function in the case of learning a structured function, only leaving output weights βi to be adjusted according to:

$$\min_{\beta} \|H\beta - T\|^2, \qquad (8)$$

The above problem is well established and known as a linear system optimization problem. Its

unique least-squares solution with minimum norm is given by

$$\beta = H^{+}T \qquad (9)$$

where **H**† is the Moore-Penrose generalized inverse of the matrix **H**. As analyzed by Bartlett [160] and Huang the generalization performance of a SLFN tends to be better with smaller magnitude of output weights. From this sense, the solution produced by ELM in Eq. (9) not only achieves the minimum square training error but also the best generalization performance on novel patterns. We now summarize ELM as the follows:

Given a training set $N = \{(\mathbf{x}_k, \mathbf{t}_k) \mid \mathbf{x}_k \in \mathbf{R}_n, \mathbf{t}_k \in \mathbf{R}_m, k = 1, \ldots, N\}$, an activation function $g(x)$, and the number of hidden neurons N

(i) Randomly assign input weights $\mathbf{w}_i$ and biases b_i according to some continuous probability density function.

(ii) Calculate the hidden layer output matrix **H**.

(iii) Calculate the output weights β_i: ˆ β = **H**†**T**.

In our experiments with ELM in this paper, the activation function is a sigmoidal function is given by

$$g(x) = 1/(1 + e{-}x)$$

And the probability density function is a uniform distribution function in the range from -1 to 1 [45].

The resolution of a general linear system $Ax = y$, where A may be singular and may even not be square, can be made very simple by the use of the Moore–Penrose generalized inverse. A matrix G of order n x m is the Moore–Penrose generalized inverse of matrix A of order m x n, if

$$\mathbf{AGA} = \mathbf{A}, \quad \mathbf{GAG} = \mathbf{G}, \quad (\mathbf{AG})^{T} = \mathbf{AG}, \quad (\mathbf{GA})^{T} = \mathbf{GA}.$$

For the sake of convenience, the Moore–Penrose generalized inverse of matrix A will be denoted by $A^{\dagger}$.

CHAPTER SEVEN

SIMULATION SETUP & RESULTS

Table 3 shows some of the parameters of our simulation experiment. MATLAB program is used to simulate the WEAC & AI assisted WEAC protocol for packet level simulation. H.263 standard parameters were used for video traffic simulation we selected QCIF frame size because of reasonable resolution quality and 56 kbps (low bit rate) bit streams after compression. Compression ratio is 10:1. The reason is that H.263 is fit for motionless video. Hence it can compress to such a high ratio. After compression, the bits are packetized into fixed packet size of 2 KB. The data rate is 3.5 packets per seconds (56 kbps). Video data is very sensitive to delay because in order for the communication to be meaningful, the data has to be received before a maximum threshold of delay 250 msec otherwise the reception of data would be of no use. Therefore, special care must be taken to minimize the delay as much as possible.

Packet Size	2Kbyte
Other Control Packet Size	100Byte
Frame Size	176x144(QCIF)
Bits per Pixel	0.2
Bit Rate	56kbps
Link Speed	5.5Mbps
Medium Access Technique	CSMA/CA
Maximum Tolerable Delay	250msec
Average Codec Power/Packet	500mW
Video Codec	H.263
Average Compression Delay	50-60msec

Table 3: Parameters for WEAC Protocol Implementation

The AI assisted WEAC routing protocol will be different from the conventional WEAC protocol as it will not be using the hello messages to update its routing table. Instead each node will predict the next locations of all the nodes within a network from the AI based prediction model and after prediction, it will find the minimum distance with its current location and new predicited location to know whether the other nodes are within in accessible range or not. If it is accessible within the range, then it will store it in its routing table. For generating the AI based prediction model, we have described the details for generating the datasets in the previous section. Table 4 shows a sample dataset for the movements of a mobile node. Fig. 12 shows the mobility pattern of a mobile node that was generated using the Gaussian Markov model within simulation grid of x and y axis in meters. Fig. 13 shows the random mobility pattern of five different mobile nodes moving in the same simulation grid. We generated different datasets of different sizes by changing the number of instances of each dataset. Datasets 1,2,3,4 & 5 have two, five, ten, fifteen and twenty thousand instances respectively.

Sub Pattern	Inputs 1	Inputs 2	Inputs 3	Inputs 4	Desired Output
1	394,138	393,138	392,138	391,138	392,138
2	393,138	392,138	391,138	392,138	392,140
3	392,138	391,138	392,138	392,140	389,140
4	391,138	392,138	392,140	389,140	386,141
5	392,138	392,140	389,140	386,141	384,141
6	392,140	389,140	386,141	384,141	381,141
7	389,140	386,141	384,141	381,141	379,141
8	386,141	384,141	381,141	379,141	378,142
9	384,141	381,141	379,141	378,142	377,142
10	381,141	379,141	378,142	377,142	376,142

Table 4: Sample dataset generated from Gaussian Markov mobility model

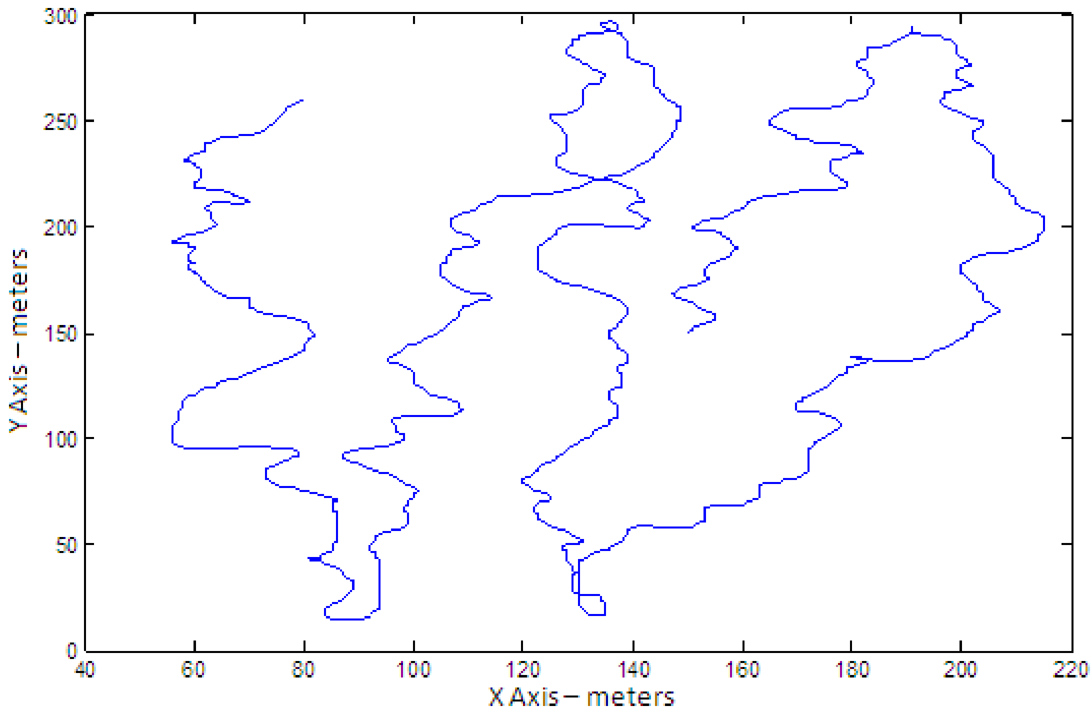


Fig. 12: Movement of a mobile node using Gaussian Markov Model with random alpha

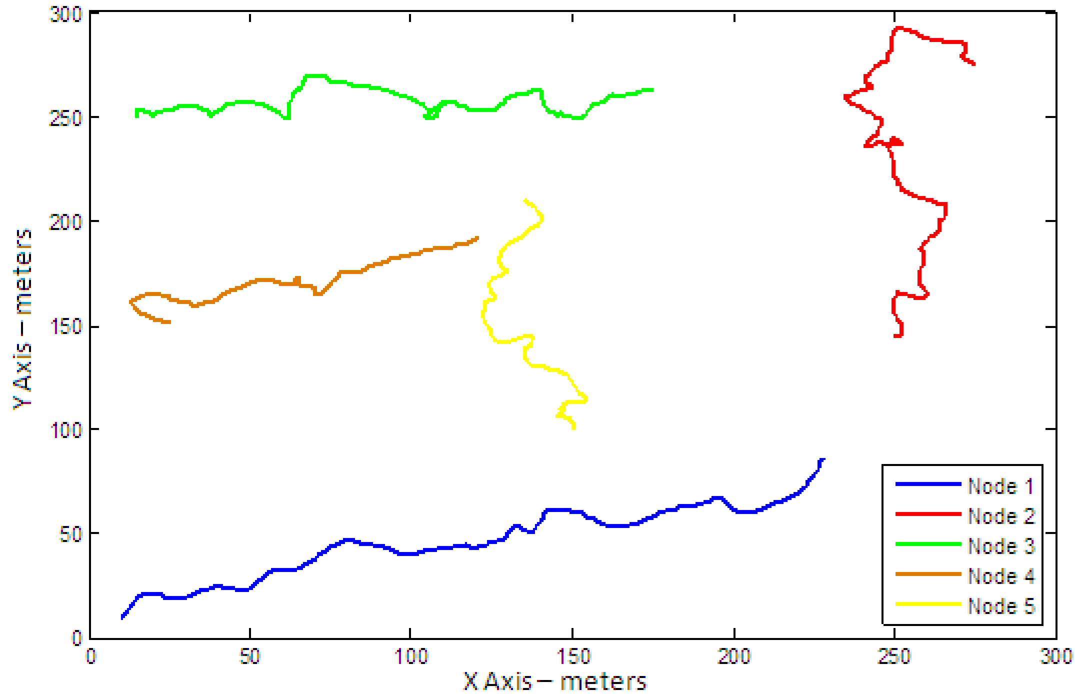


Fig. 13: Random movement of different mobile nodes within one simulation grid

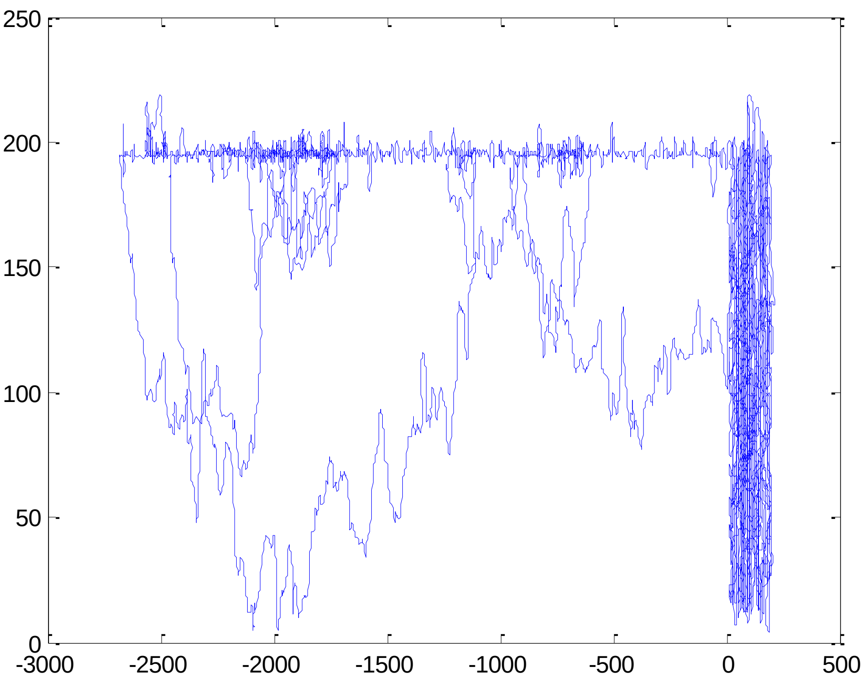


Fig. 14: Movement of a mobile node using Gaussian Markov Model with alpha = 0.1

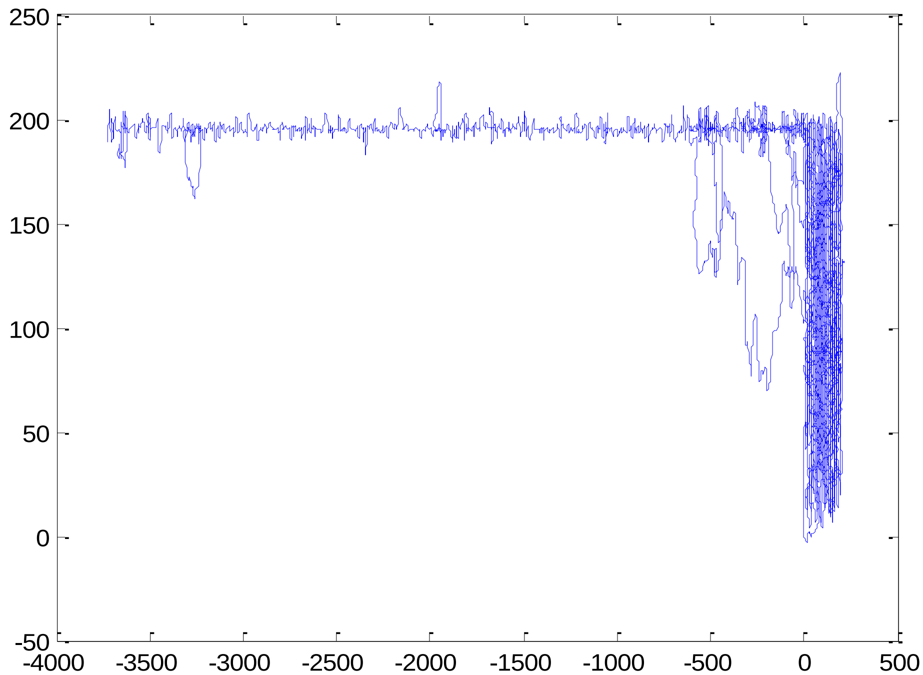


Fig. 15: Movement of a mobile node using Gaussian Markov Model with alpha 0.5

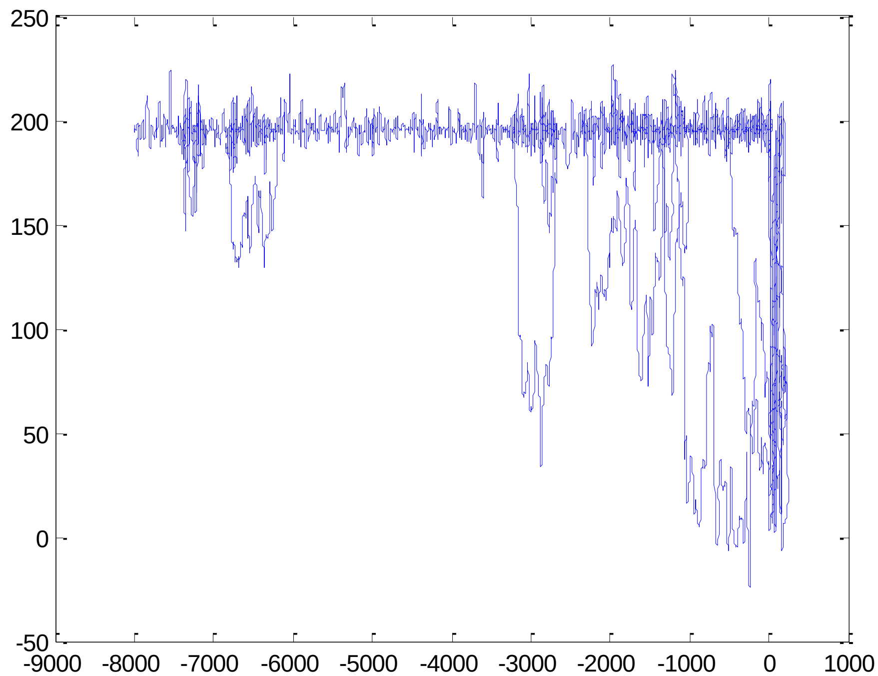


Fig. 16: Movement of a mobile node using Gaussian Markov Model with alpha = 0.9

Results with fixed alpha show the abnormal behavior of the movement of the mobile node but when the value of alpha is pure random that it is giving the accurate behavior of the mobile node movement as shown in figures 14-16. So we for building the prediction model, we will be using the one with pure random as it is used in figure 13.

7.1 Prediction Accuracy

Prediction accuracy which is the accuracy for the models will be calculated by number of times correct prediction of mobile node divided by the total number of times the prediction of mobile node in percentage. The datasets are divided into different number of instances. So for every case, 70% of data instances are reserved for training and the remaining 30% for testing. For both the testing phase, the predicted output is compared with the desired or original output to know the accuracy. Fig. 17-28 shows the mean and maximum accuracy in terms of percentage (%). The figures illustrate that the prediction accuracy is increasing as the number of instances or size of the dataset increases. It is because for both the models either neural networks or ELM, both have the same characteristics that with the increase in the number of instances, they have more and more learning or training and as the learning or training gets higher, the accuracy is also getting better. For our predication model, we are concerned about the model which gives the best accuracy. The simulation is run for one hundred times because for every iteration, it gives different prediction accuracy. For that case neural network gives the best and maximum accuracy of 30% for dataset of twenty thousand instances.

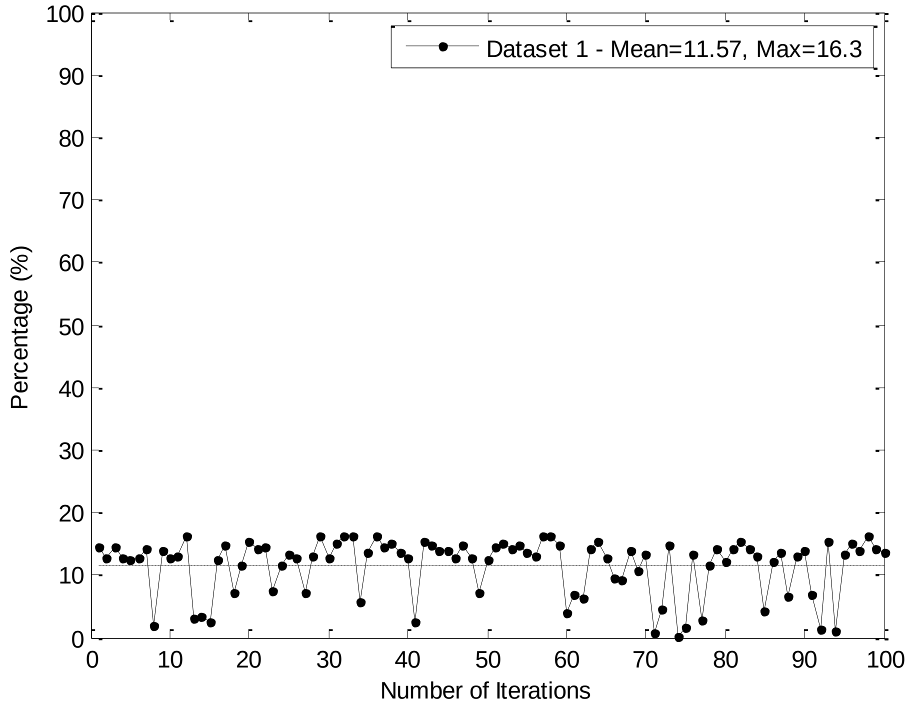


Fig. 17: Prediction Accuracy of Dataset 1 for ELM

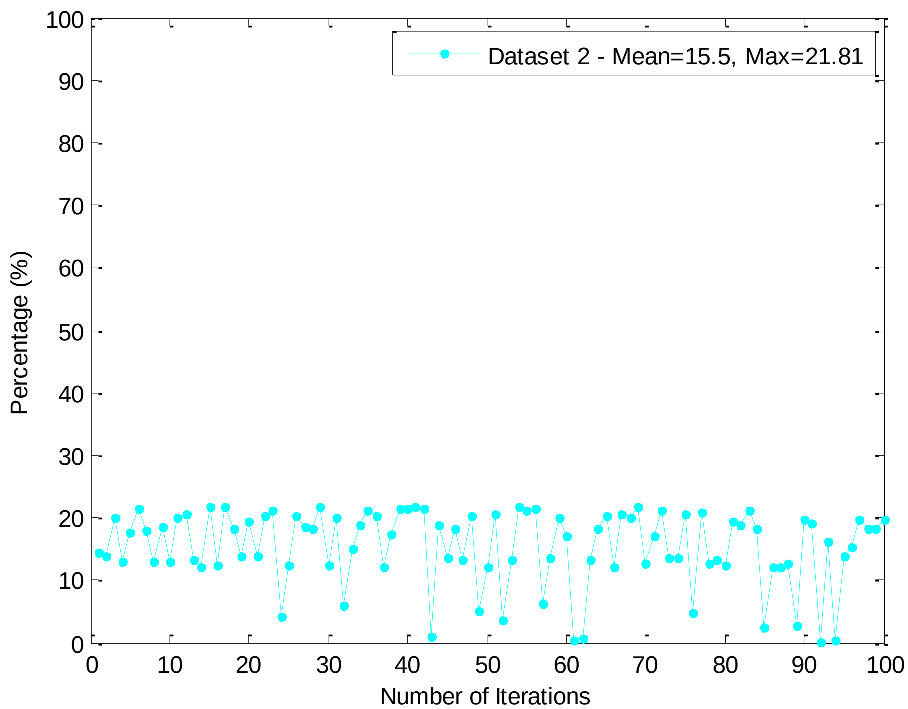


Fig. 18: Prediction Accuracy of Dataset 2 for ELM

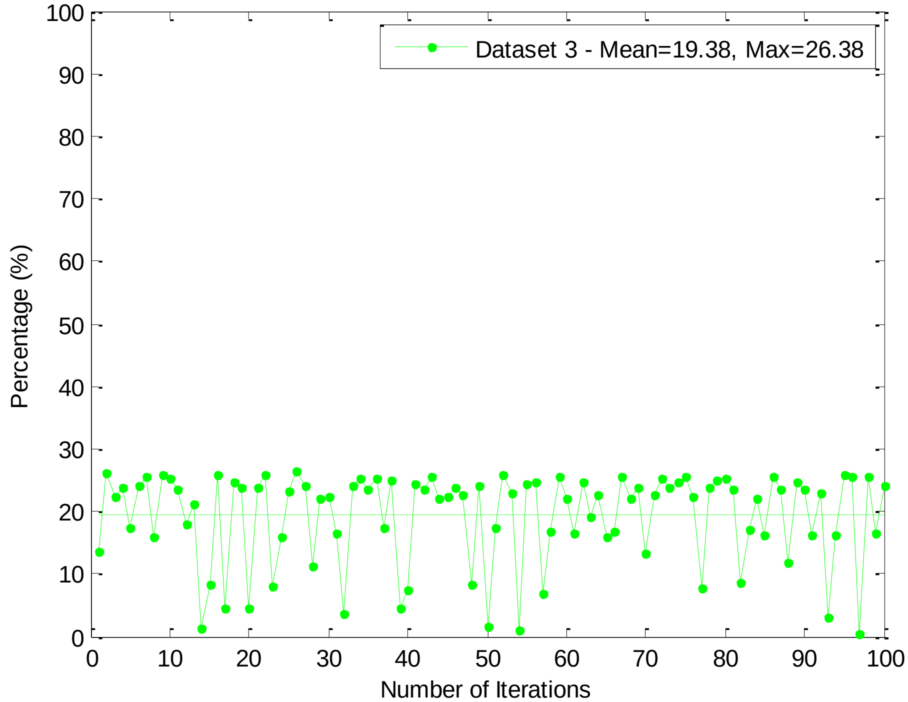


Fig. 19: Prediction Accuracy of Dataset 3 for ELM

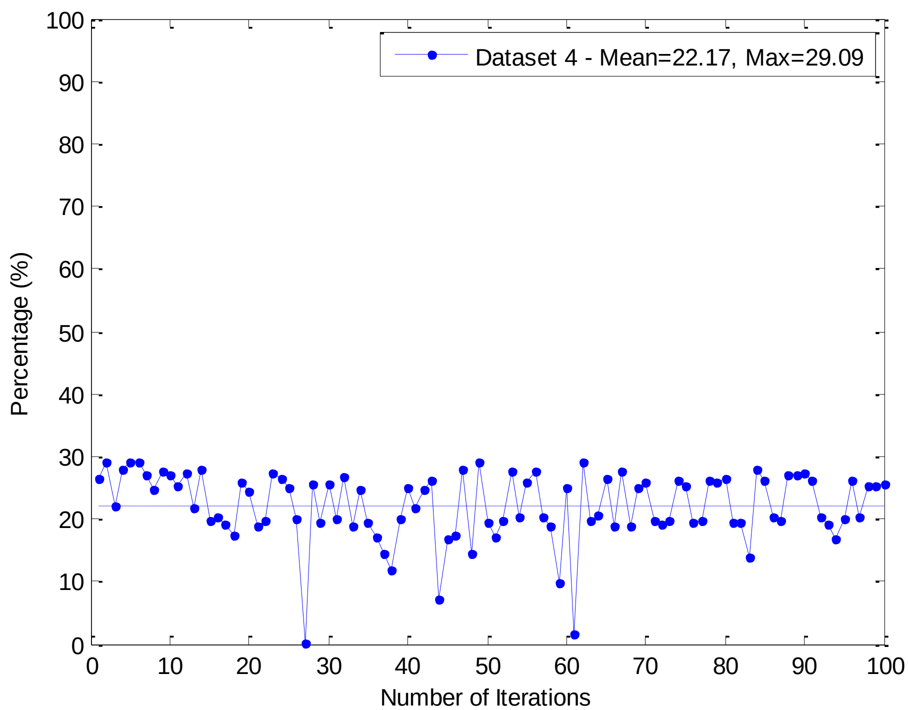


Fig. 20: Prediction Accuracy of Dataset 4 for ELM

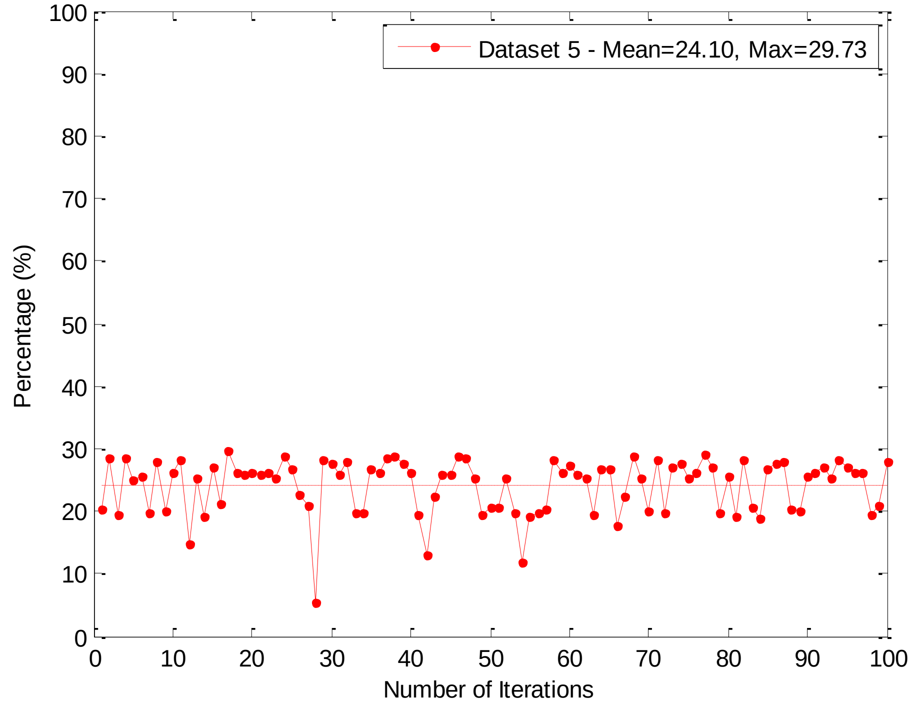


Fig. 21: Prediction Accuracy of Dataset 5 for ELM

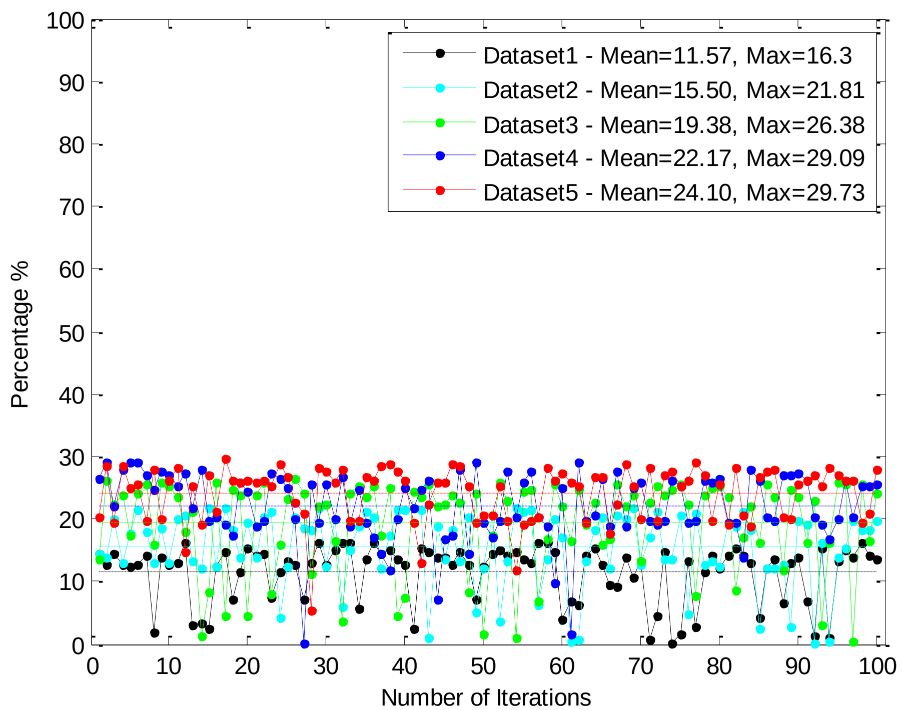


Fig. 22: Cumulative Prediction Accuracy of All Datasets for ELM

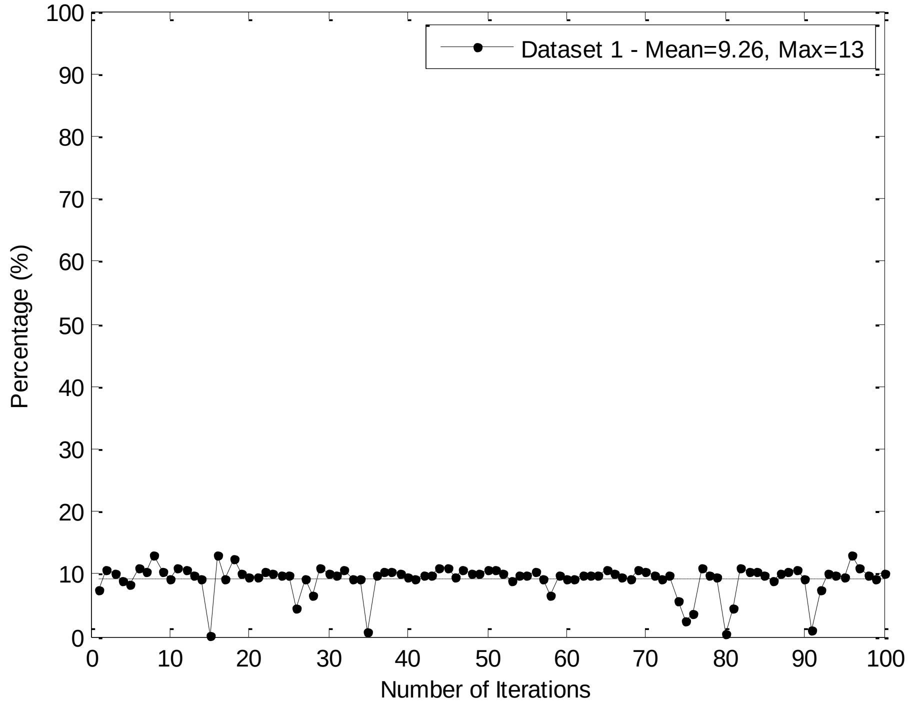


Fig. 23: Prediction Accuracy of Dataset 1 for Neural Networks

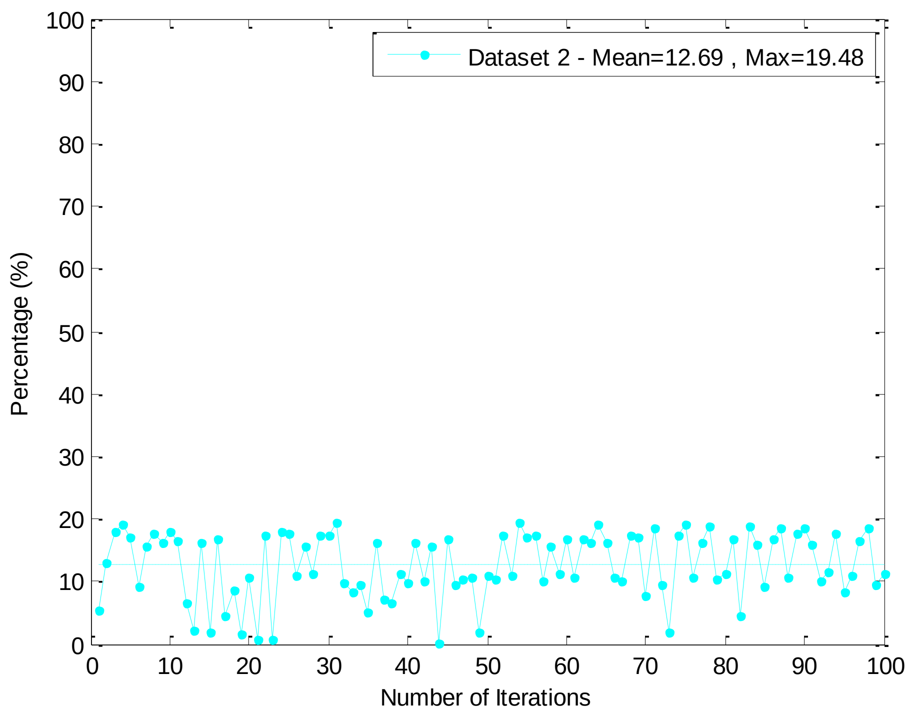


Fig. 24: Prediction Accuracy of Dataset 2 for Neural Networks

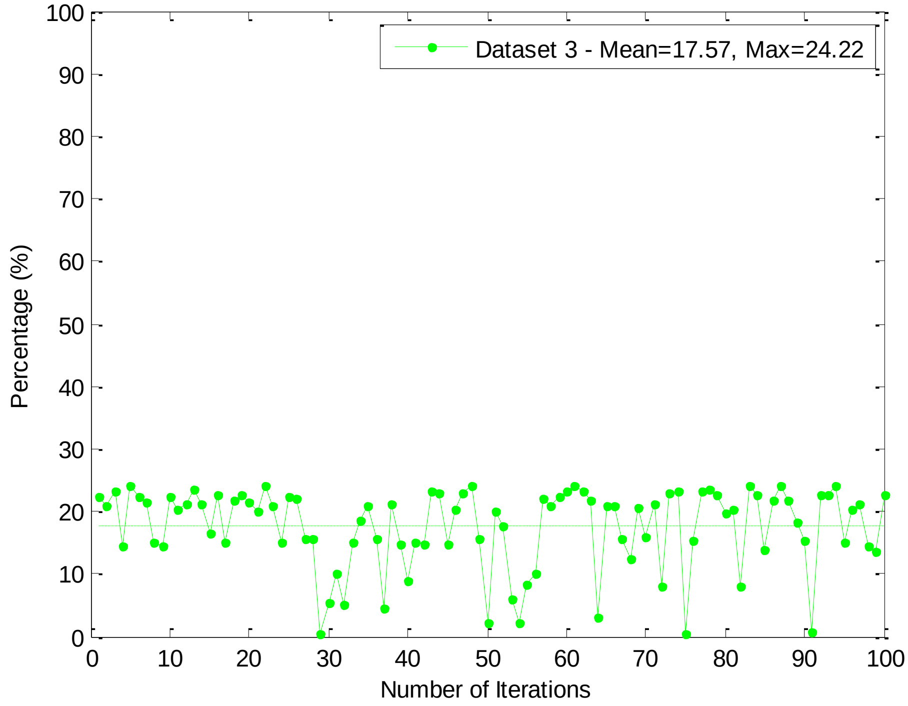


Fig. 25: Prediction Accuracy of Dataset 3 for Neural Networks

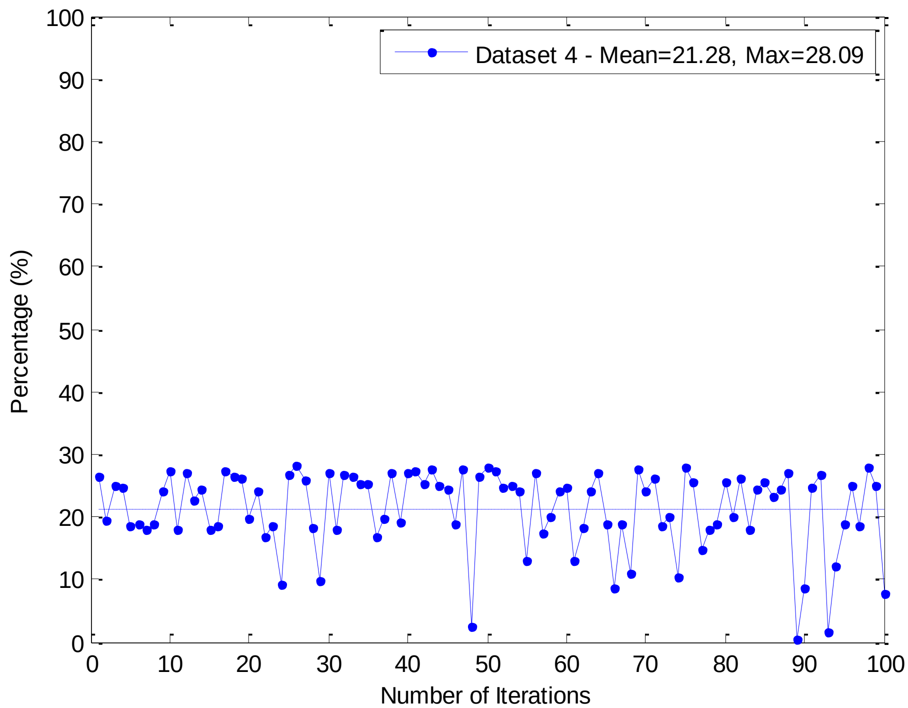


Fig. 26: Prediction Accuracy of Dataset 4 for Neural Networks

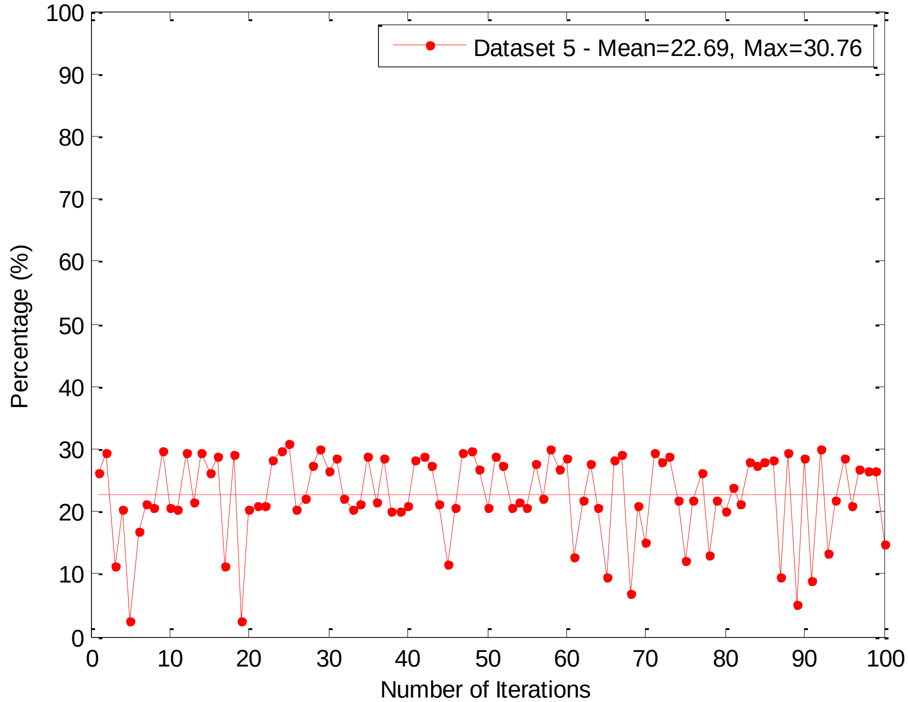


Fig. 27: Prediction Accuracy of Dataset 5 for Neural Networks

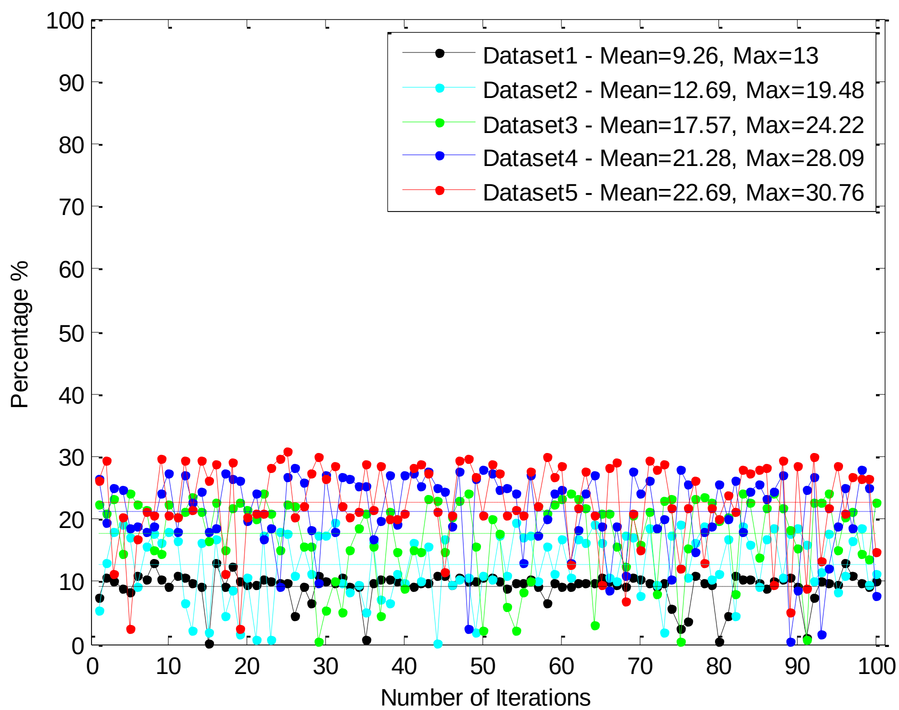


Fig. 28: Cumulative Prediction Accuracy of All Datasets for Neural Networks

Datasets	Instances	Mean (%) ELM	Max (%) ELM	Mean (%) ANN	Max (%) ANN
1	2000	11.57	16.3	9.26	13
2	5000	15.5	21.81	12.69	19.48
3	10000	19.38	26.38	17.57	24.22
4	15000	22.17	29.09	21.28	28.09
5	20000	24.1	29.73	22.69	30.76

Table 5: Mean and Max Prediction Accuracy for ELM & NN

7.2 Execution Time

The execution times of the training and testing phase for both the models whether neural networks or ELM is shown in Fig. 29 and 30. With increase in the size of dataset or number of instances, the training and testing time is also increasing. The reason for this is that the bigger the dataset, the more the learning or training time the model will take. The training time for neural network is more than that of ELM but we are not concerned much about the training time because for our prediction model, the training will be done offline, only the trained model will be used to give us the next location. So for that case, testing time is the important factor that is needed to be considered. Keeping this thing in mind, the neural network is giving better accuracy of 30% over ELM which is giving the prediction accuracy of 29%. For our prediction model, we will be using ELM even though it's not better in comparison with neural network in terms of accuracy but at the same time, its testing time is much better than that of neural networks which will have huge impact on our routing algorithm in terms of total delay. So, the model which is giving 3.0 sec testing time with 70% of twenty thousand instances will be used in the simulation.

Datasets	**Instances**	**Training Time (sec)**	**Testing Time (sec)**
Dataset 1	2000	7.6	2.3
Dataset 2	5000	9	2.9
Dataset 3	10000	12.8	3.5
Dataset 4	15000	19	4.6
Dataset 5	20000	27	5.2

Table 6: Execution Time for ELM

Datasets	**Instances**	**Training Time (sec)**	**Testing Time (sec)**
Dataset 1	2000	2.9	1.49
Dataset 2	5000	3.2	1.68
Dataset 3	10000	5	2.01
Dataset 4	15000	8.2	2.38
Dataset 5	20000	12.1	3

Table 7: Execution Time for ANN

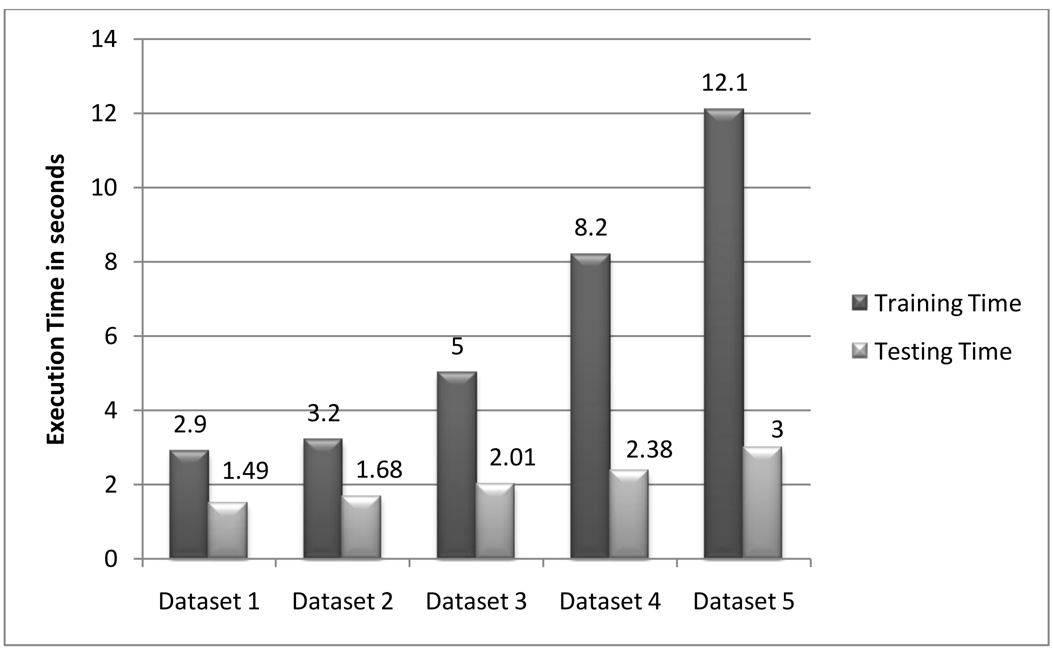


Fig. 29: Execution time of all datasets for ELM

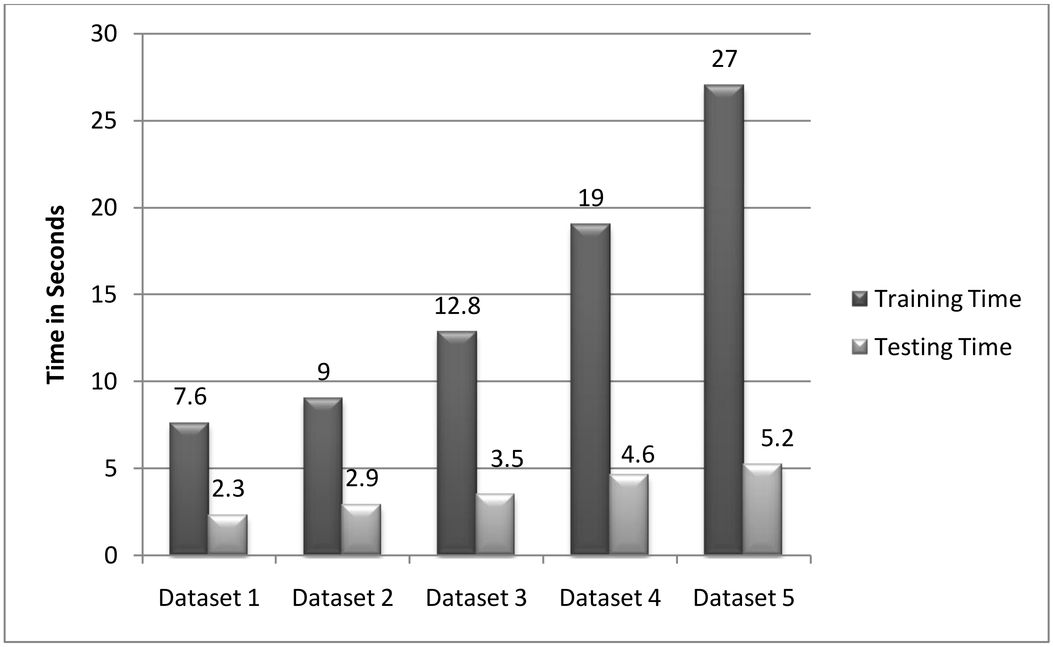


Fig. 30: Execution time of all datasets for Neural Networks

7.3 Average Packet Delay

In our experiment, compression / decompression delay, routing delay, CSMA/CA delay, propagation delay, transmission delay and queuing delay constitute the overall delay.

Propagation Delay = d / Link Speed (where d is the distance between two nodes)

Transmission Delay per Packet = Packet Size / Data Rate

Fig. 31 shows the plot of delay for different cases that are dealing with conventional and AI based routing. With the AI assisted WEAC protocol, each prediction will take 0.0005 msec to predict the next route as per the execution time of ELM. For the one that is using the conventional routing that is variation in average delay as we increase the number of nodes. It varies somewhat between 100 msec and 320 msec, which is almost within the acceptable range of 250 msec until N exceeds 100. There are a few clusters and the packet has less number of available routes to the destination, which result in either less routing delay or more packet drop. This is why it is difficult to predict the behavior of the network before reasonable connectivity because sometimes there is less routing delay if the packet obtains desired connectivity but at times the failure to obtain desired connectivity results in packet drop and hence larger overall delay. However, as the network gets congested with a large number of nodes, there will be more cluster formation and hence more available paths for a packet to reach its final destination. The other scenarios that are using the proposed AI technique for routing the packets, the behavior is same as that of conventional routing but it is having more delays in comparison with the conventional routing. The reason for this is that the AI model to predict the next location of building the routing tables and updating the neighbor list is taking more time which is having an impact of delay. For another case, we assume that AI prediction delay is assumed to be zero, so for that case, it is performing much better in comparison with conventional routing which also shows that AI prediction is useful for the cases where the AI computation delay is less or approximately equal to zero.

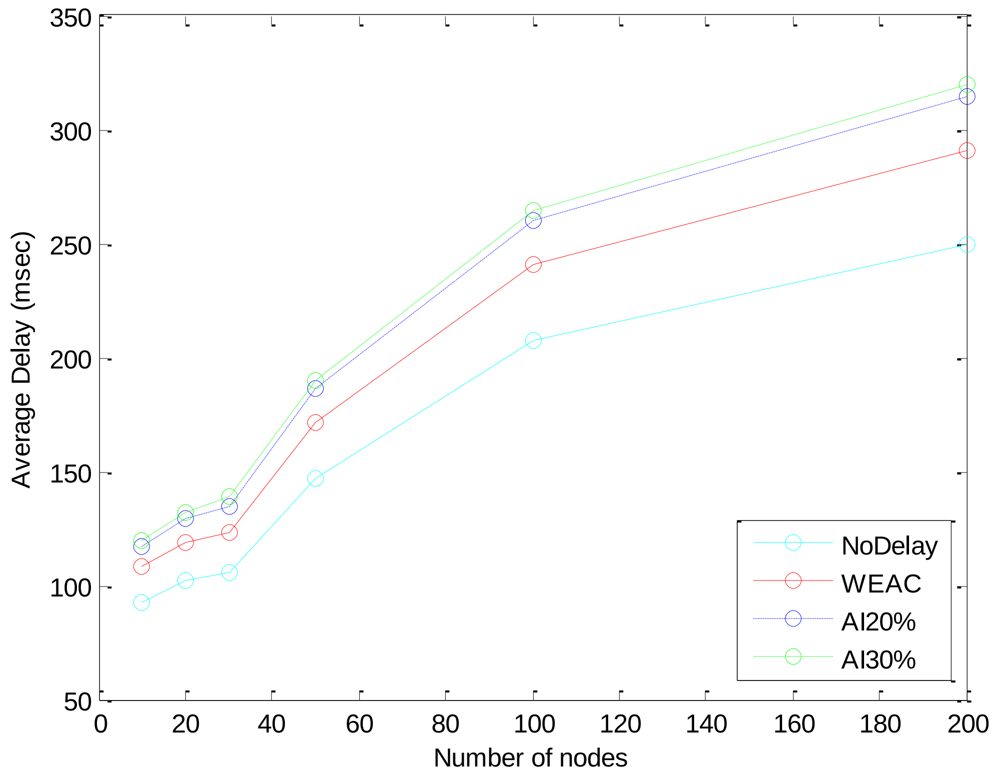


Fig. 31: Average Delay for different routing techniques.

7.4 Average Successful Transmission

Fig. 32 shows that poor connectivity is obvious when the number of nodes is less than 30 or so especially if they stay far away from each other. This behavior has been shown in Fig. 32 for nodes less than 50. But as the number of nodes reaches 50, we see the highest percentage of successfully received packets because of maximum connectivity. This percentage again decreases for larger number of nodes because the network gets congested with more number of control packets which results in higher queuing delay and more number of packet loss. Moreover, access to the medium also becomes difficult with such a high number of nodes. For the case when we are using the AI prediction model for routing, the successful transmissions are low in comparison with the conventional routing. This is because the AI prediction model that we have used is giving the prediction accuracy of 30% which eventually results in more packet loss. So, the better the prediction accuracy better will be the successful transmission and this is quite obvious from the plot when we used the prediction models with accuracy of 20% and 30%.

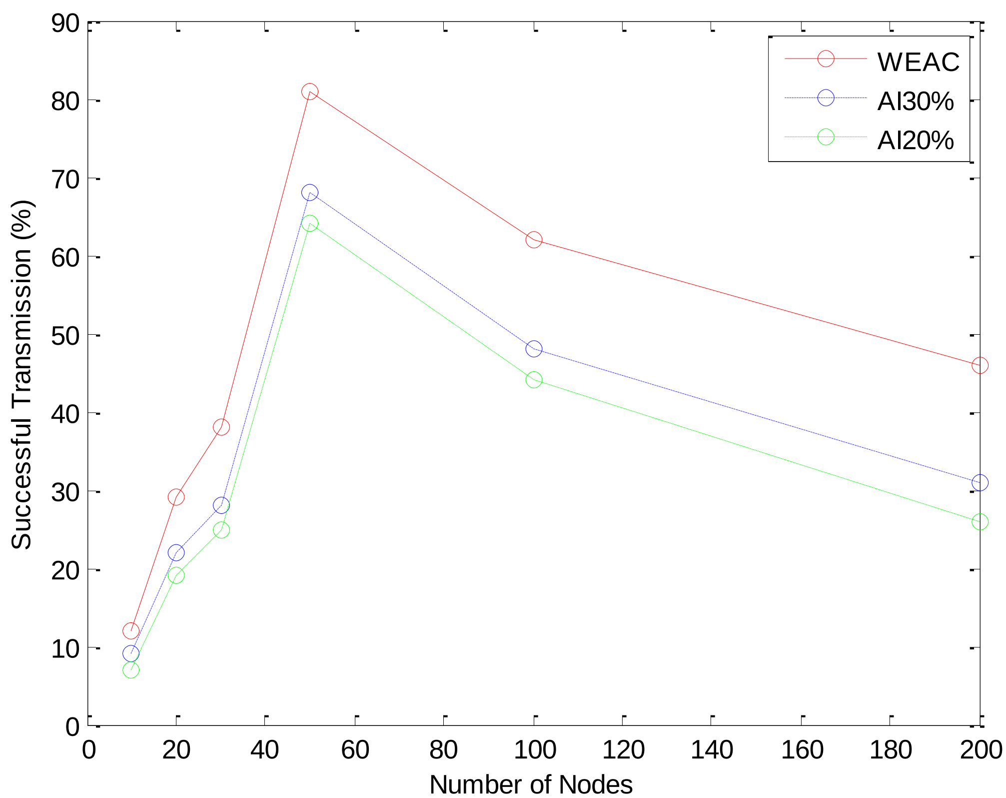


Fig. 32: Average successful transmission for different routing techniques

7.5 Energy – Transmission Power

Since mobile devices rely on battery power, it is important to minimize their energy consumption which can be significant, especially for smaller devices. The multi-hop routing problem in ad-hoc networks has been widely studied in terms of bandwidth utilization, but energy consumption has received less attention. It is sometimes wrongly assumed that bandwidth utilization and energy consumption are roughly the same. In this case, we will study the total energy costs associated with a packet containing some number of bytes of data. Ad-hoc mode operation does not use any base station. Nodes communicate directly with all other nodes that are in wireless transmission range. Because there are no base stations to moderate communication, hosts must always be ready to receive traffic from their neighbors. In the simple case, the energy consumed by the network interface when a host sends and receives or a packet can be described using a linear equation

$$Energy_{send} = m_{send} \times size + b_{send} \text{ [76]}.$$

$$Energy_{receive} = m_{receive} \times size + b_{receive} \text{ [77]}.$$

Where ‘*m*’ & ‘*b*’ represent fixed cost as shown in Table 10.

Parameters	WLAN (Send)	WLAN (Receive)
m	0.000405	0.000157
b	0.07894	0.04209

Table 8: Linear models for the sending and receiving costs

The model also does not consider energy consumed in unsuccessful attempts to acquire the channel or messages lost due to collision, bit error or loss of wireless connectivity. Such effects are difficult to obtain for controlled experimental measurements [76] [77]. Fig. 33 shows that there is a significant saving in terms of energy from both cases either sending or receiving when it is compared for both the routing protocols either conventional or routing using AI model.

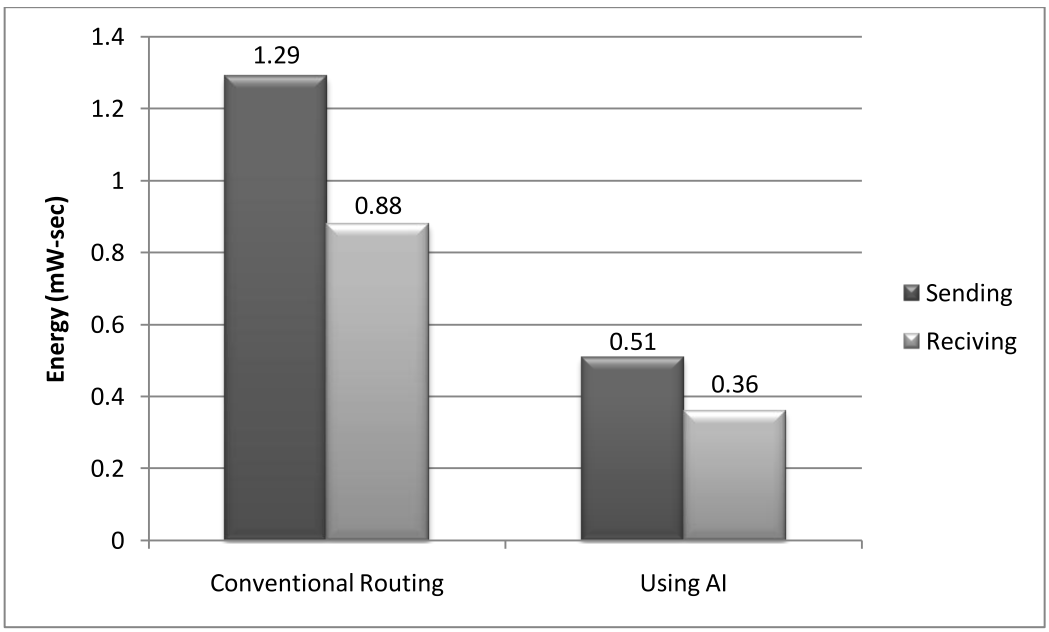


Fig. 33: Costs for sending and receiving for different routing techniques.

CHAPTER 8

CONCLUSION & FUTURE WORK

This thesis emphasizes on the replacing the conventional routing for ad-hoc networks to AI based prediction model that predicts the next location of the mobile nodes and updates the routing table. The approach uses two different AI techniques, multilayer neural network and ELM to predict the future location of a mobile host based on the history of movement pattern of a mobile host. The performance of the method has been verified for prediction accuracy by considering different movement patterns of a mobile host and learning accuracy. As there is no standard dataset available to train the dataset, we implemented Gaussian Markov model to have some realistic random movements. These different datasets were used to train the models with different size of instances. Simulation is also carried out for different movement patterns to predict the future location of a mobile host for one hundred times. The results show that with higher the size of dataset, more prediction accuracy was achieved up to data instances of twenty thousand samples. ELM and Neural Networks were giving accuracy of 29% & 30% respectively.

The execution time for both the models were calculated and results showed that ELM much faster convergence as compared to neural networks. So ELM based prediction model was used for simulating the WEAC protocol. The results conveyed that the AI is having more delay in comparison with conventional routing due to the delay of AI computation but for the scenario where we used AI computation delay to be zero, the delay reduced meaningfully and giving much better performance against the conventional routing. For the successful packet transmissions of AI model is having lower rates because of less prediction accuracy which results in more packet loss. In terms of energy, the routing algorithm with AI consumes less energy than that of conventional delay. So for the future work, more AI techniques which include SVM, hidden markov models, adaptive networks should be investigated to increase the robustness of the prediction framework and which should have less computation time and more prediction accuracy. This will eventually has great impact on delay & packet loss.

References

[1] Milena Stoyanova and Petri Mähönen "A Next-Move Prediction Algorithm for Implementation of Selective Reservation Concept in Wireless Networks", IEEE, Personal, Indoor and Mobile Radio Communications, 3-7 Sept. 2007.

[2] G. Karayannis, "Emerging Wireless Standards: Understanding the Role of IEEE802.15.4 & ZigBee in AMR & Submetering," 2003, http://www.zigbee.org/imwp/idms/popups/pop_download.asp?contentID=820. Last accessed on 21 April 2005.

[3] K. Intae and R. Poovendran, "Maximizing static network lifetime of wireless broadcast ad-hoc networks ", in Proceedings of the IEEE International Conference on Communications, 11-15 May 2003, Anchorage, USA , vol. 3, 2003, pp. 2256 – 2261.

[4] W. Stallings, Data & Computer Communications , 6th ed., Prentice Hall, New Jersey, p. 453, 2000.

[5] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," Computer Networks (Elsevier), vol. 38, pp. 393-422, 2002.

[6] Deepak Ganesan, Ramesh Govindan, Scott Shenker and Deborah Estrin Highly-Resilient, Energy-Efficient Multipath Routing in Wireless Sensor Networks ACM Mobile Computing and Communications Review, Vol. 5, No. 4, October 2001.

[7] J.-H. Chang and L. Tassiulas, Maximum Lifetime Routing in Wireless Sensor Networks, Proc. Advanced Telecommunications and Information Distribution Research Program (ATIRP2000), College Park, MD, Mar. 2000.

[8] Rahul C. Shah and Jan Rabaey, Energy Aware Routing for Low Energy Ad-hoc Sensor Networks IEEE Wireless Communications and Networking Conference (WCNC), March 17-21, 2002, Orlando, FL.

[9] T. Camp, J. Boleng, and V. Davies. A Survey of Mobility Models for Ad-hoc Network Research. Wireless Communications & Mobile Computing (WCMC): Special issue on Mobile Ad-hoc Networking: Research, Trends and Applications, 2(5):483--502, 2002.

[10] B.P. Vijay Kumar and P. Venkataram " Prediction-based locationmanagement using multilayer neural networks", Journal of Indian institute of science, Vol. 82, No. 1, pp.7-22, Jan-Feb 2002.

[11] A. Bar-Noy, I. Kessler, and M. Sidi. Mobile users: To update or not to update? In Proceedings of the Join Conference of the IEEE Computer and Communications Societies (INFOCOM), pages 570–576, 1994.

[12] J.J. Garcia-Luna-Aceves and E.L. Madrga. A multicast routing protocol for ad-hoc networks. In Proceedings of the Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), pages 784–792, 1999.

[13] M. Sanchez and P. Manzoni. Anejos: A java based simulator for ad-hoc networks. Future Generation Computer Systems, 17(5):573–583, 2001.

[14] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, A performance comparison of multi-hop wireless ad-hoc network routing protocols, in Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking(Mobicom98), ACM, October 1998.

[15] J. Broch, D. Maltz, D. Johnson, Y. Hu, and J. Jetcheva. Multi-hop wireless ad-hoc network routing protocols. In Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM), pages 85–97, 1998.

[16] D. Johnson and D. Maltz. Dynamic source routing in ad-hoc wireless networks. In T. Imelinsky and H. Korth, editors, Mobile Computing, pages 153–181. Kluwer Academic Publishers, 1996.

[17] E. Royer, P.M. Melliar-Smith, and L. Moser. An analysis of the optimum node density for ad-hoc mobile networks. In Proceedings of the IEEE International Conference on Communications (ICC), 2001.

[18] Z. Haas. A new routing protocol for reconfigurable wireless networks. In Proceedings of the IEEE International Conference on Universal Personal Communications (ICUPC), pages 562–565, Oct. 1997.

[19] B. Liang and Z. Haas. Predictive distance-based mobility management for PCS networks. In Proceedings of the Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), March 1999.

[20] B. L. Lu, M. Ichikawa, and S. Hosoett. A Modular Massively Parallel Learning Framework for Brain-Like Computers. In IEEE International Conference on Systems, Man, and Cybernetics, volume 5, pages 332–337, 1999

[21] E. Daprati and A. Sirigu. How we interact with objects: learning from brain lessons. TRENDS in Cognitive Sciences, 10(6), june 2006.

[22] M. Arbib. The Handbook of Brain Theory and Neural Networks. MIT Press, 1998.

[23] G. P. Zhang. Neural Networks for Classification: A Survey. In IEEE Transactions on Systems Man and Cybernetics-Part C, volume 30, pages 451–462. 2000.

[24] Bishop, C.: Neural Network for Pattern Recognition. Clarendon Press, Oxford (1995)

[25] Neural Network Applications. http://www.ip-atlas.com/pub/nap/.

[26] K. K. Ang and C. Quek. Stock Trading Using RSPOP: A Novel Rough Set-Based Neuro-Fuzzy Approach. IEEE Transactions on Neural Networks, 17(5):1301 – 1315, 2006.

[27] R.Lawrence. Neural Networks to Forecast Stock Market Prices http://people.ok.ubc.ca/rlawrenc/research/Papers/nn.pdf. 1997.

[28] H. Kordylewski, D. Graupe, and K. Liu. A novel large-memory neural network as an aid in medical diagnosis applications . Information Technology in Biomedicine, IEEE, 5(3):202 – 209, September 2001.

[29] C. S. Pattichis, C. N. Schizas, and L. T. Middleton. Neural network models in EMG diagnosis. Biomedical Engineering, IEEE, 42(5):486 – 496, May 1995.

[30] M. Stevens, C. Anderson, and J. R. Beveridge. Efficient Indexing for Object Recognition Using Large Networks. In IEEE International Conference on Neural Networks, June 1997.

[31] B. L. Zhang, H. Zhang, and S. S. Ge. Face recognition by applying wavelet subband representation and kernel associative memory. In IEEE Transactions on Neural Networks, volume 15, pages 166 – 177, Jan 2004.

[32] A. Zidouri, M. Sarfraz, S. A. Shahab, and S. S. Jafri. Adaptive Dissection Based Sub word Segmentation of Printed Arabic Text. In IEEE International Conference on Information Visualization (IV'2005), pages 991–996, London, UK, 2005.IEEE Computer Society Press.

[33] D. Lowe and K. Zapart. Validation of neural networks in automotive engine calibration. In Fifth International Conference on Artificial Neural Networks, volume 440, pages 221 – 226, July 1997.

[34] S. Haykin. Neural Networks: A Comprehensive Foundation. Prentice Hall, 2nd edition, July 1998.

[35] C. Bishop. Neural Networks for Pattern Recognition. Oxford University Press, USA, November 1995.

[36] S. J Russel and P. Norvig. Artificial Intelligence, A Modern Approach. Prentice Hall, 2nd edition, 2003.

[37] H. Demuth and M. Beale. Neural Network Toolbox http://www.mathworks.com. 2004.

[38] J. A. Leonard and M. A. Kramer. Radial-basis function networks for classifying process faults. IEEE Control Systems Magazine, 31(8), 1991.

[39] M. D. Buhmann and M. J. Ablowitz. Radial basis functions: Theory and implementations. Cambridge University Press, 2003.

[40] S. Haykin. Neural Networks: A Comprehensive Foundation. Prentice Hall, 2nd edition, July 1998.

[41] G.-B. Huang, Q.-Y. Zhu and C.-K. Siew, Extreme learning machine: A new learning scheme of feedforward neural networks, in *Proc. IEEE Int. Joint Conf. Neural Networks* **2** (2004) (Budapest, Hungary) 985–990.

[42] P. L. Bartlett, The sample complexity of pattern classification with neural networks: The size of the weights is more important than the size of the network, *IEEE Transactions on Information Theory* **44** (1998) 525–536.

[43] http://www3.ntu.edu.sg/home/egbhuang/

[44] E. B. Baum, On the capabilities of multilayer perceptrons, *J. Complexity* **4** (1988) 193–215.

[45] G.-B. Huang, Q.-Y. Zhu and C.-K. Siew, Extreme learning machine: Theory and applications, accepted in *Neurocomputing*, (2006).

[46] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance vector routing (DSDV) for mobile computers," in Proceedings of the conference on Communications architectures, protocols and applications SIGCOMM '94, vol. 24, no. 4, October 1994.

[47] S. Murthy and J. J. Garcia-Luna-Aceves, "A routing protocol for packet radio networks," in Proceedings of the 1st annual international conference on Mobile computing and networking. ACM Press, December 1995.

[48] C. C. Chiang, "Routing in clustered multihop, mobile wireless networks with fading channel," Proceedings of IEEE SICON 97, pp. 197–211, April 1997.

[49] J. J. Garcia-Luna-Aceves and M. Spohn, "Source-tree routing in wireless networks," in Proceedings of the Seventh Annual International Conference on Network Protocols. IEEE Computer Society Washington, DC, USA, 1999, p. 273.

[50] G. Pei, M. Gerla, and T.-W. Chen, "Fisheye state routing in mobile ad-hoc networks," in ICDCS Workshop on Wireless Networks and Mobile Computing, 2000.

[51] C.-K. Toh, Ad-hoc Mobile Wireless Networks: Protocols and Systems. Prentice Hall Engineering, December 2001.

[52] R. D. et al., "Signal stability based adaptive routing (SSA) for ad-hoc mobile networks," IEEE Personal Communication, vol. 4, no. 1, pp. 36–45, February 1997.

[53] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad-hoc wireless networks," in Mobile Computing, Kulwer, 1996, pp. 153–181.

[54] V. D. Park and M. S. Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks," in Proc. INFOCOM 97, vol. 3, April 1997, pp. 1405–1413.

[29] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in Second IEEE Workshop on Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA '99., February 1999, pp. 90–100.

[56] Z. J. Haas, M. R. Pearlm, and prince Samar, The Zone Routing Protocol (ZRP) for ad-hoc networks, IETF MANET Internet Draft, July 2002.

[57] C. Santivanez, S. Ramanathan, and I. Stravrakakis, "Making link state routing scale for ad-hoc networks," in Proceedings of the 2nd ACM international symposium on Mobile ad-hoc networking and computing. ACM Press, October 2001, pp. 22–32.

[58] V. Ramasubramanian, Z. J. Haas, and E. G. Sirer, "Sharp hybrid adaptive routing protocol (SHARP)," in Proceedings of the 4th ACM international symposium on Mobile ad-hoc networking and computing. ACM Press, June 2003, pp. 303– 314.

[59] T. Sheltami and H. Mouftah, “An efficient energy aware clusterhead formation infrastructure protocol for MANETs,” in 8th IEEE International Symposium on Computers and Communication, vol. 1, 2003, pp. 203–208.

[60] 4i2i Communications, “H.263 video coding tutorial,” available at http://www.4i2i.com/h263 video codec.htm, 2004.

[61] M. Oliver and A. Escudero, “Study of different CSMA/CA IEEE 802.11-based implementations,” in EUNICE 1999 Contribution, 1999.

[62] A. H. Sadka, Compressed Video Communications. John Wiley and Sons, Ltd, April 2002.

[63] K. Ozan, Tonguz, and G. Ferrari, AD-HOC WIRELESS NETWORKS. John Wiley and Sons, Ltd, 2006.

[64] S. Mao, S. Lin, P. S.S., Y. Wang, and E. Celebi, “Video transport over ad-hoc networks: Multistream coding with multipath transport,” in IEEE Journal of Selected Areas in Communications, ser. 10, vol. 21, December 2003, pp. 1721–1737.

[65] S. Mao, D. Bushmitch, S. Narayanan, and S. S. Panwar, “MRTP: A multi-flow real-time transport protocol for ad-hoc networks,” in Proceedings of 58th IEEE Vehicular Technology Conference, vol. 4, October 2003, pp. 2629–2634.

[66] S. Mao, S. Lin, S. S. Panwar, and Y. Wang, “A multipath video streaming testbed for ad-hoc networks,” in Proceedings of the Fall 58th IEEE Vehicular Technology Conference, vol. 5, October 2003, pp. 2961–2965.

[67] S. Mao, S. Lin, S. Panwar, and Y. Wang, “Reliable transmission of video over ad-hoc networks using automatic repeat request and multi-path transport,” in Proceedings of 54th IEEE Vehicular Technology Conference, vol. 2, October 2001, pp. 615–619.

[68] S. Lin, S.Mao, Y.Wang, and S. Panwar, “A reference picture selection scheme for video transmission over ad-hoc networks using multiple paths,” in Proceedings of the 2001 IEEE International Conference on Multimedia and Expo, August 2001,pp. 96–99.

[69] Y. Wang and S. Lin, “Error resilient video coding using multiple description motion compensation,” in IEEE Transaction on Circuits and Systems on Video Technology, vol. 12, June 2002, pp. 438–452.

[70] E. Masala, C. Chiasserini, M. Meo, and J. D. Martin., “Real-time transmission of H.264 video over 802.11-based wireless ad-hoc networks,” in publications of CERCOM.

[71] S. Lin and Y. Wang, “Analysis and improvement of multiple description motion compensation video coding for lossy packet networks,” in International Conference on Image Processing, vol. 2, September 2002, pp. 185–188.

[72] T. K., U. N., and S. Y., “Optimal data rate control for video stream transmission over wireless network,” in 18th International Conference on Advanced Information Networking and Applications (AINA), vol. 1, 2004, pp. 340–345.

[74] S. Buruhanudeen, “Wireless ad-hoc networks,” available at http://www.comp.brad.ac.uk/ sburuha1/wirelessadhoc.htm, November 2002.

[73] T. Yoo, E. Setton, X. Zhu, A. Goldsmith, and B. Girod, “Cross-layer design for video streaming over wireless ad-hoc networks,” in IEEE 6th Workshop on Multimedia Signal Processing, 29 September 2004, pp. 99–102.

[74] M. A. Weiss, Data Structures and Algorithm Analysis in C++, 3rd ed. Addison Wesley, 2006.

[75] Qun Li and Javed Aslam and Daniela Rus. Hierarchical Power-aware Routing in Sensor Networks In Proceedings of the DIMACS Workshop on Pervasive Networking, May, 2001.

[76] P. Gauthier, D. Harada, and M. Stemm, “Reducing power consumption for the next generation of PDAs: It’s in the network interface!,” research report, University of California at Berkeley, Berkeley, California, Jan 1996.

[77] L.M. Feeney, M. Nilsson, Investigating the energy consumption of a wireless network interface in an ad-hoc networking environment, in: Proceedings of INFOCOM 2001.

[78] http://en.wikipedia.org/wiki/List_of_ad-hoc_routing_protocols

VITAE

- Mohammad Rahil Rafiq
- Born in Karachi, Oct 2nd, 1983
- Received Bachelor of Science (B.S) in Computer Engineering from Sir Syed University of Engineering & Technology, Karachi, Pakistan in March 2006.
- Joined King Fahd University of Petroleum & Minerals, Dhahran, Saudi Arabia as a Research Assistant in February 2007.
- Completed Master of Science (M.S.) in Computer Networks in May 2010.
- Email: rahil.rafiq@gmail.com
- Present Address: P.O.BOX 1472, Dhahran – 31261, Saudi Arabia.
- Permanent Address: Flat No. 5, WS-8, Marium Apartment, Faran Housing Society, Karachi, 74800.